JAN/FEB '87
Vol. 3 No. 2
$3.00 U.S. Funds

BULK RATE
U.S. POSTAGE
PAID
COLTON, OR 97017
PERMIT #51

POSTMASTER:
PLEASE EXPEDITE DELIVERY
DATED MATERIAL
ADDRESS CORRECTION REQUESTED

AMERICA'S LARGEST TIMEX SINCLAIR MAGAZINE

TIME DESIGN MAGAZINE

WHAT'S NEW FOR

1 9 8 7

COMPLETE SINCLAIR FORECASTS

PLUS...

RLE Graphics • Tax Calculator

TASWORD TWO +

Build a Data Acquisition Module

Exploring the QL • QL Quill

AND MORE!

SALE!★

640K QL.... $ 399!*

QL, with 4 Psion programs, 4 blank mdv cartridges, and 640K RAM $399.00+
512K RAM PAK for QL, brings total memory to 640K..... $199 PP!
AVATEX 300/1200 BAUD MODEMS, 8 LED, RS232, etc..... $109 PP!
AVATEX 300/1200, 100% Hayes Compatible..... $159 PP!
QL MODAPTER, allows use of 300 Baud modem with QL, includes QCODE terminal program which supports up/downloading, etc..... $ 59.95+
Modapter WITH AVATEX 1200 HC (Hayes Compatible)..... $209 PP!
QL ASSEMBLER WORKBENCH, includes monitor, disassembler, assembler, by Talent, a great buy, great software! Only..... $ 49.95+
TRANSFORM CASE, plastic case holds 20 MDV cartridges..... $ 5.99+
MICRODRIVE CARTRIDGES for QL, ZX Microdrives ONLY(4 carts).. $ 10.95+
QL BOOKS, Assorted, call for titles, only..... $ 9.95+

THE 2068 is NOT forgotten !!

$$$ SALE $$$>>> Fighter Pilot, Night Gunner, or Loader V, only $10 each POST PAID while current supply lasts!

SPRITES 2068, all new! Design, combine, and use sprites quickly and easily. Includes GREAT demos! Includes detailed manual, sprite service utility (2520 bytes MC), spritedraw program..... $ 19.95+

SABOTEUR, you are a ninja spy who must get a disk from the enemy wharehouse; use lethal kicks, punches, stars, knives, grenades to 'remove' guards and dogs who 'get in the way'. Super graphics! $ 15.95+

CRITICAL MASS, aliens threaten to destroy your solar system by taking over your anti-matter conversion plant. You must infiltrate the plant and destroy the converter before BOOM! and it turns into a black hole, destroying all stars and planets in your system..... $ 15.95+

SPECTERM 64- Brand new 64 column terminal software for the Spectrum or 2068 with Spectrum Emulator and 2050 modem OR ANY RS232 BOARD! Yes, 1200 BAUD (300 baud with 2050 modem) on a 2068! You get 64 columns ON SCREEN, built in XModem up and downloading, half/full duplex, 31K buffer, can transmit/receive ANY file- Bytes or Program. Automatically saves Tasword II files properly! 4K user menu area, and documentation for full power modeming, all for only..... $ 29.95+

DOUBLE SIDE/DOUBLE DENSITY floppy disks, highest quality, lifetime guarantee, Tyvec sleeves, user ID labels, and write protect tabs, only 75 CENTS each (minimum 10), or 100 for 65 cents each! (100 shipped PP!)

75¢ DISKS

Unless otherwise noted, please add $3 shipping for software, $5 for hardware, or $10 for a QL package.

Send check or Money Order to:

VARIETY SALES COMPUTER DIVISION
Department TDN-3
325 WEST JERSEY STREET, #2D
ELIZABETH, NEW JERSEY 07202
201-527-0535 FOR MORE INFORMATION.

LETTERS

JAN/FEB '87 VOL. 3 NO. 2

TIME DESIGNS MAGAZINE

FOR ALL TIMEX AND SINCLAIR COMPUTERS

TIME DESIGNS MAGAZINE CO.
29722 Hult Rd. • Colton, Oregon 97017

(503) 824-2658
CompuServe ID# 71350,3230

TIME DESIGNS MAGAZINE is published bi-monthly and is Copyright © 1986 by the **Time Designs Magazine Company,** Colton, Oregon 97017. All rights reserved.

Editor: Tim Woods

Assistant Editor: Stephanie Woods

Editorial Assistant/Production: D.L. Woods

Photography:
(unless otherwise noted): Thomas Judd
Printing by; Toad'l Litho Printing and Comp., Oregon City, Oregon 97045

SUBSCRIPTIONS: $15 a year for six issues (US funds only). No extra charge to Canadian subscribers. All other countries please write for information on air mail rates.

CUSTOMER SERVICE: Customer satisfaction is our goal. For subscription service problems please write or call TIME DESIGNS.

CHANGE OF ADDRESS: Write or call to prevent delay of sevice,

Reproduction of this magazine in whole or in part by any means without written permission is prohibited by law

**NOTICE: Contributors to TIME DESIGNS are independent of the TIME DESIGNS MAGAZINE CO., and opinions expressed in the contents of the magazine are not necessarily those of the management or its advertisers. Time Designs Magazine Co. will not be held liable for any damage or consequences resulting from instructions, assertions of fact, review of products or companies provided in the magazine's content."*

Renew Your Subscription Today!

Micro-Prolog

Dear Sir,

Reference the question in your last magazine about the availability of the program "Prologue" for the ZX Spectrum computer. Sinclair Research commissioned a version of the Prolog language for the Spectrum and sold it for about 30 Brit. pounds. However, when the rights to the Spectrum were transferred to Amstrad, all production ceased. Morever, Amstrad refuses to publish the Prolog software. So much for the bad news, now the good news...a copy of "Micro-Prolog" is available from: Logic Sales Ltd., 6 Midgate, Peterborough, Cambs. PE1 1TN, England (limited quantities available).

The cost for this program is £3.95 + £ 4.00(airmail postage); total £7.95. The pound is currently worth about $1.45(U.S.). Anyone who wants a copy had better hurry; there are only a few copies left. The reason that the airmail postage is £4.00 is that the cassette comes in a presentation case with a 24 page introduction booklet, plus a 301 page primer on Micro-Prolog.

Sincerely,

Charles R. Byler
Ft. Riley, KS

A & J help request

To the Editor & Readers,

I have a major problem (and others may have this same problem) with "self-starting" programs. This involves trying to put these programs on to an A&J Micro Drive for the TS-2068.

Does anyone have a program that can be pre-loaded into the computer then merged with the self-starting program, so that the Save instructions can be changed to allow it to go to the A&J instead of the cassette recorder? Sure hope someone out there can help.

Thanks kindly,

Vince Stimmel
Hendersonville, NC

Editor: While I heartily endorse and support the A&J as a low-cost mass storage device, I found that it's user guide is very skimpy. We regularly receive all kinds of requests for help with this system. I do hope that others will share info and tidbits for the A&J. Is there such a routine...similar to an "007" or "Spy" routine?

2068 Music

Program Contributed By:
Joe E. Jenkins
Amarillo, TX

```
  10 FOR z=1 TO 18: READ a,b,c
  20 SOUND 7,60;8,15;9,15;0,a;1,
b;2,a+2;3,b
  30 PAUSE 10*c: SOUND 8,0;9,0:
NEXT z: STOP
  40 DATA 92,4,2,209,5,2,92,4,2,
209,5,2,92,4,1,158,4,1,47,5,1,15
8,4,1,92,4,4,226,3,2,209,5,2,226
,3,2,209,5,2,226,3,1,209,5,1,47,
5,1,209,5,1,92,4,4
```

Letters continued on page three...

NEW PRODUCTS

For TS2068 and SPECTRUM

Artworx

VERSION 1.1

- Pull-Down Menus
- Several Brushes
- Spray
- Auto-Fill
- Zoom
- Undo
- Several Text Fonts
- Cut & Paste Windows
- Auto-Speed Control
- Magnify & Reduce
- Rotate & Mirror
- Full Attribute Control
- Fully Elastic Shapes including Circle, Box, Triangle, Ray and Line
- Fast Ellipse and Arc
- Includes Spectrum & TS2068 Versions
- Supports Microdrives and Kempston Joystick
- Includes GALLERY, the slide show/animator
- 5 Samples of Artwork
- Excellent Manual

Unshackle your creativity with ... ARTWORX!!

ARTWORX V1.1 - $19.95 U.S. plus $3.00 S & H

REVOLUTIONARY NEW BASIC COMPILER...

TIMACHINE

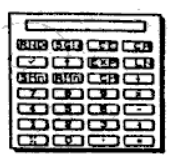
The dream of every BASIC programmer has now been realized!

- TIMACHINE will turn your BASIC into super-fast machine code, running up to 200 times faster!
- Handles floating point operations like SIN, COS, TAN
- Handles all BASIC except I/O
- Includes an excellent manual and 4 demonstration programs
- Compiles up to 27K in seconds
- Includes Spectrum & 2068 versions

Super-Charge your BASIC programs with...*TIMACHINE!*

TIMACHINE — $19.95 U.S. Plus $3.00 S & H

THE WORX!



A unique combination of planning aids, decision tools, and utilities.

- Scheduler/Planner
- Notepad
- Telephone Book
- Programmable Calculator
- Decision Factoring
- Real Time Clock
- Biorhythms
- Perpetual Calendar
- International Time Zones
- Superb Manual
- Includes Spectrum & TS2068 Versions
- Supports Microdrives

A TIMELY ADDITION TO YOUR SOFTWARE LIBRARY...THE WORX!

— $19.95 U.S. Plus $3.00 S & H

NovelSoft

A FORMAT FOR THE FUTURE

106 Seventh Street, Toronto, Ontario, Canada M8V 3B4 • TEL. (416)259-8682 • CompuServe 70416,1435

L E T T E R S

MSCRIPT

First of all, I would like to encourage all of the suppliers (magazines, newsletters, developers of software and hardware) to continue offering useful software or modifications to existing software. I suspect that the majority of Timex computerist are not all that interested in learning the "ins and outs" of programming, hacking, and hardware development. We just want something practical which we can use on a day to day basis. Please don't think I am knocking those who have a more techinical interest. But neither do I wish to be criticized for my lack of technical interest.

As examples of the above comment, I offer MSCRIPT (especially Jack Dohany's customized version CMS-5), Tasword Two, Omnicalc 2, Pro/File 2068. There are more of course but these represent the ones most useful to me.

Second, a word about MSCRIPT. I have seen almost nothing about this tremendous program. I used WordStar for quite a long time on my brother's computer and I can honestly say that MSCRIPT has some distinct advantages over WordStar. Admittedly, it is not perfect but it is VERY GOOD!

I have not always felt this way about MSCRIPT. It wasn't until I saw a small note in T-S Horizons about Jack Dohany's "Jack's Fairware". The article mentioned a customized version of MSCRIPT which I immediately sent for. This is the best piece of software that has been written for the 2068 computer. The original version of MSCRIPT, and Tasword Two are nothing by comparison. In fact, the only deficiency I have found is that end of page markers are not available (as in WordStar). You have to print the document to see where the pages break. Tasword Two does not do this either.

I would like to see Jack get the recognition and credit he deserves for this fine word processor. I use it several times a week, every week.

Looking forward to my next issue of Time Designs.

Richard Templeton
Springfield, MO

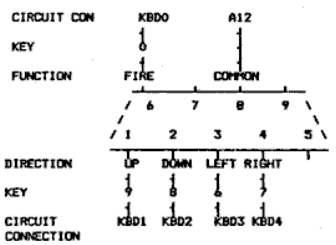
Editor: While giving Jack proper recognition, I might mention that he has a new address- Jack Dohany, 390 Rutherford Ave., Redwood City, CA 94061. Folks who send a legal size S.A.S.E. can get complete details on the Customized MSCRIPT Vers.5 and other "fairware".

Sinclair Joysticks

In regard to the recent query from a Time Designs reader about using the Kempston protocol joystick interface by modifying the program or use of user-definable keys (for reference- see the article "Adding a Joystick To The Spectrum-ized 2068" in May/June '86 issue of TDM); there is no easy way to do that short of modifying the program itself. However, there is one simplistic joystick interface which is "cheap and dirty" (but it works), which can be used with both Sinclair joystick interface and keyboard options.

Very simply, the Sinclair interface mimics the keyboard keys 6,7,8,9 and 0 for joystick control. This can be done by simply wiring a joystick in directly, or providing a socket for attachment. The following diagram indicates the wiring as viewed from the wiring side of a DB9 socket. This type of interface would have to be the low cost leader as far as joystick interfaces go.

Jack Keene
Dallas, TX



80 Col. VU-CALC

Dear Tim,

Here are some program lines for people with Timex's VU-CALC and an 80 column printer.

After calculating and saving your calculations to tape or disk, break out of VU-CALC (Enter #Q then Option #1). Merge your printer interface software (it can't occupy addresses 34573 to 52072). Add these lines:

```
9700 FOR P=0 TO 280 STEP 70
9710 FOR Z=34573 TO 52072 STEP 350
9720 FOR L=0 TO 69
9730 LPRINT  CHR$ ( PEEK (Z+P+L));
9740 NEXT L
9745 LPRINT
9750 NEXT Z
9755 FOR T=0 TO 30: LPRINT : NEXT T
9760 NEXT P
```

A "GOTO 9700" will print 5 pages of VU-CALC (the total worksheet, not just the screen copy "window").

To keep 7 digit numbers from running together, add these lines (this also separates every 7th letter if you have TEXT):

```
9735 IF (L+1)/7= INT ((L+1)/7) THEN LPRINT " "; (1 space)
DELETE line 9745
```

If you do not use the workspace below line "AN" and want a neat looking double spaced work sheet, change these lines:

```
9710 FOR Z=34573 TO 48500 STEP 350
9745 LPRINT : LPRINT
DELETE line 9755
```

If you only want one page, change these lines:

```
page 1 - column 1  to 10 : 9700 LET P=0
page 2 - column 11 to 20 : 9700 LET P=70
page 3 - column 21 to 30 : 9700 LET P=140
page 4 - column 31 to 40 : 9700 LET P=210
page 5 - column 41 to 50 : 9700 LET P=280
                           DELETE line 9760
```

A "GOTO 3000" will give you the option to return to VU-CALC.

Carl Green
East Liverpool, OH

spinning wheel

This short program for the TS2068 draws several different "views" of a wheel, then "spins" the wheel for an interesting animated effect. Try it out!

Program Contributed by:
Charles Goyette
Drummondville, Quebec
Canada

```
   4 REM If you don't want to se
e the wheel being drawn , insert
 90 PAPER 7: INK 7
 134 INK 0
   5 CLEAR 29999
  15 LET a=30000
  20 READ q
  25 POKE a,q
  30 LET a=a+1: IF a<30047 THEN
GO TO 20
  35 DATA 33,0,64,17,0,120,1,0,2
4,237,176,201,1,0,40,197,62,5,33
,0,120,0,17,0,64,1,0,24,237,176,
6,20,14,220,13,32,253,16,249,61,
32,235,0,193,16,225,201
 100 GO TO 900
 110 LET l=USR 30000
 115 LET r=PEEK 30004+256*PEEK 3
0005+6144
 120 POKE 30005,INT (r/256)
 125 POKE 30004,r-256*PEEK 30005
 130 NEXT f
 135 CLS
 140 REM Machine Routine Starts
Here
 145 LET l=USR 30012
 150 STOP
 900 LET q=0
 910 FOR f=1 TO 5: CLS : CIRCLE
126,85,80: CIRCLE 125+(q/3),84+(
q/3),83: CIRCLE 126,85,10
 920 FOR t=0+q TO 95+q STEP 5
 930 LET a=t/50*PI
 940 LET sx=126+79*SIN a
 950 LET sy=85+79*COS a
 960 PLOT 126,85: DRAW sx-125,sy
-85
 970 NEXT t
 980 LET q=q+1
9999 GO TO 110
```

Character Font Generator

Now, you can design your own UDGs and character sets with this great programmer's tool. It even comes with 4 fonts to get you started. Plus, it can make sprites for use with the Advanced Video Modes. One joystick required.

$20.00 US $25.00 Can

Advanced Video Modes

The most complete package for utilizing all of the T/S 2068's enhanced screen displays to their fullest. It provides access to 32, 64, and even 80 columns, plus dual screen and extended colour modes. Updates are available for just $8.00 Can/US.

$15.00 US $20.00 Can

Send cheque or money order for **$1.50** US/Can. to receive our catalogue complete with **Demonstration Tape V3.0** plus bonus programs. And remember, we **still** pay the postage & handling charges on all our orders!

Beaver Computer Products

756 Fleming Avenue, Winnipeg, Manitoba, R2K 1V5

'The best DAM software around'

Now at last . . . TS-2068

The FootePrint Printer Interface

- for Centronics parallel printers
- works in both 2068 and Spectrum mode
- compatible with OS-64 & Spectrum emulators
- EPROM socket and on/off switch on board
- works with **both** Tasman and Aerco driver software
- plugs into cartridge dock—door completely closes with cable running back under computer
- frees up rear edge connector allowing other peripherals to be used; less chance of a crash
- print driver software for LPRINT, LLIST, and COPY included for 2068 and Spectrum modes

FootePrint Interface w/software & cable .$45.00
FootePrint with OS-64 option included . .$65.00
Bare board & instructions only $15.00
Cable only for use with bare board $15.00
All prices are pre-paid and include shipping charges.

FOOTE SOFTWARE
P. O. Box 14655 — Gainesville, FL 32604
904/462-1086 (6 pm - 9 pm EDT)

HAPPY NEW YEAR!
FROM
RMG ENTERPRISES
****

TO START 1987 OUT ON THE RIGHT FOOT, WE JUST WANT TO REMIND YOU THAT WE HAVE A BRAND NEW 40+ PAGE 8 1/2"x11" CATALOG JUST WAITING FOR YOU TO BROWSE THROUGH!

IF YOU HAVE NOT ALREADY ORDERED YOUR COPY, HURRY AND DO SO! WE KNOW THAT YOU WILL FIND THINGS IN IT THAT YOU MAY NOT HAVE EVEN KNOWN EXISTED! THINGS THAT YOU NEED!

WE NOW STOCK OVER 550 DIFFERENT ITEMS THAT ARE USABLE DIRECTLY OR INDIRECTLY WITH THE SINCLAIR LINE OF COMPUTERS.

SEND FOR YOURS TODAY!
JUST SEND A CHECK OR MONEY ORDER FOR $3.00 TO THE ADDRESS BELOW, AND WE WILL RUSH YOUR BIG NEW CATALOG TO YOU! INSIDE YOU WILL FIND A LOT OF HARDWARE AND SOFTWARE THAT MAY SUIT YOUR NEEDS. AND YOU WILL ALSO FIND A COUPON GOOD FOR $3 OFF ON YOUR FIRST ORDER FROM THE CATALOG! THAT WAY YOU DON'T LOSE A THING!

DETAILS ARE ALSO INCLUDED ON HOW TO KEEP THIS CATALOG UP-TO-DATE AND CURRENT. AS IT IS THE LAST FULL CATALOG RMG WILL EVER PUBLISH, THIS WILL BE IMPORTANT TO YOU!

SEND TO:
RMG ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OR 97045
503/ 655-7484
TUE-SAT 10 AM TO 10 PM

New and Upcoming in 1987

By Tim Woods and D. Hutchinson

INTRODUCTION: You've seen those glaring tabloids on the news rack at your local grocery store...you know, the ones boasting new trends and predictions for the new year...all hyphenated in bold print and splashed accross their front page. While we won't comment on the accuracy or validity of these publications (after all, how do we know that Sylvester Stallone won't be visited by the ghost of Elvis?!?)...instead we offer our version of the same concept, but based more on "fact" and gleaned from interviews, telephone conversations and press releases.

We are going to take a "sneak preview" of what we think will be the most important events and trends for our Timex Sinclair computing community in 1987. Some of the issues we will address along the way will be--"what new hardware and software is being developed?"--"will Uncle Clive's computers survive in the U.S., now three years after Timex dropped out of the market?"--"what are some new ways I can put my computer to work for me?" and much more.

So, hold on to your hat, as we start our whirlwind tour of "what's new and upcoming in 1987"...and as our tabloid counterparts might say..."a look at what's hot, and what's not!"

Indianapolis: The Place To Be In May

If you read the reports about the Mid-West Timex Sinclair Computerfest which was held in Cincinnati, Ohio last year...you may have wished that you had been there. Well, now you have a second chance. It's time to start saving your nickels, dimes, and "cash in" on your pop bottles. That's right, the 2nd Annual Mid-West Timex Sinclair Computerfest will be held in Indianapolis, Indiana on May 2nd and 3rd...and plans are going "full steam ahead". If you can fly, drive, ride the bus, ride the train, walk (?)...no matter what mode of transportation you use...you might not want to miss perhaps the single most important event for Sinclair enthusiasts in North America. In fact, the only event of it's kind held in the U.S. [there have only been two other such conventions: Boston in 1983 and Cincinnati in 1986].

The 1987 Mid-West Timex Sinclair Computerfest will be located at the Holiday Inn North in Indianapolis (just off North I-465 at 3850 Depauw Blvd.). This is a "Holi-dome", and one of the top hotels in the Holiday Inn chain. The exhibition area is twice as large as last year, with over 5,000 square feet of display space plus areas for seminars, conferences and a hospitality suite.

This year's executive director of the Computerfest is Paul Holmgren, who not only is a fan of Sinclair computers, but also has ties to the city commerce department of Indianapolis. Paul will be working closely with Computerfest founder and promoter, Frank Davis. Other Computerfest committe members are representatives of TS user groups from Indiana and Ohio.

The list of dealers that have indicated they will be attending the show at this early date has been impressive. These include: Sharp's Inc., C.W. Associates, Knighted Computers, Thomas B. Woods and Syncware News, Variety Sales, Zebra Systems Inc., Markel Enterprises, Time Designs Magazine Company, and The WJDJUP Company; along with several user groups from Indiana, Ohio, New York, Maryland, and Wisconsin. As we went to press, more dealers (including some of the largest TS mail order houses) expressed interest in the show.

HOLIDAY INN North

MIDWEST TIMEX SINCLAIR COMPUTER FEST

Tickets for the TS Computerfest can be purchased in advance for a discount before March 30 ($4 for an individual; $7 for the family plan)...or at the door for $6/individual or $9/family plan. The Computerfest will be open on Saturday May 2nd from 9-6pm and on Sunday May 3rd from 9-5pm. Accomodations can be obtained at the "headquarters" hotel for a special rate by mentioning the TS Computerfest (phone number is 317-872-9790); or lodging is also available at six other hotels in the immediate area. Transportation to and from the International Airport is being provided by the Holiday Inn, for those staying at the headquarters hotel.

Some of the activities planned for the Computerfest include workshops by Timex Sinclair "experts", and most of the workshops will be repeated twice, to allow for a more convenient schedule for all attendees. There will be a swap meet for used equipment or programs; and like last year's show, valuable door prizes will be given away (some lucky fellow went home with a complete QL system including computer, RGB monitor and printer!). On Friday evening at 7:00pm on May 1st, will be a banquet for all exhibitors participating in the show.

Every Timex and Sinclair computer will be featured in one way or another. The March/April '87 issue of Time Designs will carry more information on the Mid-West TS Computerfest as well as an updated exhibitor listing and schedule of events.

For more information on the show, lodging, tickets, and display booth reservations...write or call: Paul Holmgren, 5231 Wilton Wood Ct., Indianapolis, IN 46254, tel. (317) 291-6002; or Frank Davis, 513 E. Main St., Peru, IN 46970, tel. (317) 473-8030.

RAM, RAM and more RAM

Running out of places to store your data with 16k or 48k RAM? Good news for you on the horizon...and most likely, the most popular hardware addition this coming year...memory boards.

* Larry Kenny of Larken Electronics (RR#2 Navan, Ontario, Canada K4B 1H9) is currently developing a 256k RAMdisk for the TS2068.

* Another company (who asked to remain anonomous) is also working on a 256k RAM expansion board for the 2068. The operating system software is reported to be similar to Timex's proposed plans for the 2068, and will utilize commands found on the top row of keys (OPEN#, MOVE, CAT, ect.). The board will also be expandable for additional RAM.

* Thomas B. Woods will be offering a RAM board in kit form that operates in the cartridge slot of the 2068. The kit and board is designed by Tom Bent, and features user expandable RAM from 8k to 120k, with battery back-up. The kit is similar to a board that is being marketed by The E. Arthur Brown Company. For more information and prices, write to: PO Box 64, Jefferson, NH 03583.

Continued Next Page.

* In light of Wes Brzozowski's five-part article "The Mystery Of The Missing 253" on bank-switching memory for the TS2068 (to be concluded next issue), several bank-switching "spin-offs" are in the works, including some future hardware projects recently submitted to Time Designs. We think research and development in bank-switched RAM will produce some valuable "add-ons" later this year.

* Bill Pederson of The WJDJUP Company reported that he will be conducting live demonstrations of his bank-switching boards at the upcoming Mid-West TS Computerfest in Indianapolis (for further details on peripherals and software write to: 1120 Merrifield S.E., Grand Rapids, MI 49507).

* After Tim Stoddard completed the two-part article on upgrading the TS1016 RAM Pack to 64k, in TDM (see the Sept/Oct '86 and Nov/Dec '86 issues); he came up with an excellent INTERNAL 64k upgrade for the TS1000 and ZX81. It's a three chip modification, that makes a neat and tidy installation, with a substantial boost in "power" for all TS/ZX hackers. Complete plans will appear in the May/June '87 issue of Time Designs.

TS Telecommunications

Telecommunications is not necessarily a new field for TS users...as the Byte Back Modem cropped up early on in hey-days of the ZX81, then followed by the Westridge TS2050 modem, the original Timex modem. But recent developments in this area, are making telecommunications a growing support base for Timex Sinclair owners.

A new booklet tells it all. It's called "The Guide To T/S Telecommunications", written by Pete Fischer and Steve Ishii. Over 50 pages of information offer a tutorial on "how to get started", necessary equipment and software, Timex Sinclair BBS phone numbers and info about each one, and even sections for the QL. The booklet is available for $2.00 postage paid from Pete Fischer (PO Box 2002, Tempe, AZ 85281). There is also a DELUXE edition which will be available shortly in an expanded format for $5.00 postage paid. Anyone remotely interested in this area should get a copy...it could even save you money in the long run with all of the special tips, and helpful information it provides.

Many users have discovered the Timex/Sinclair area offered by the huge telecomputing information service, CompuServe. Several TS dealers have subscription packets for sale, or check at your local computer store. Price for this "starter package" is around $30. The Timex Sinclair (and QL) area can be found in the Computer Club Forum sponsored by Family Computing Magazine. Type GO CLUB at the user prompt, and your there. You will find lots of messages, programs and articles to "download", a live conference on Wendsday nights...all just for TS users...and hosted by SYSOP's Patrick Spera and Dave Rothman.

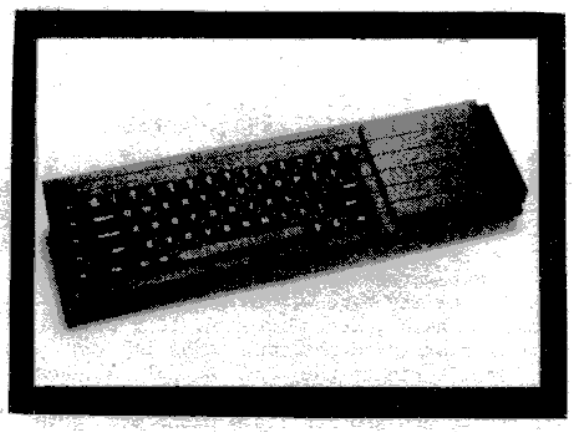
Front Cover of "The Guide To T/S Telecommunications" by Pete Fischer and Steve Ishii (the RLE graphic picture was downloaded from CompuServe).

It wasn't until just recently, that software to allow setting up ones own home BBS using a TS2068 became available. CASBOARD 2068 is available from The E. Arthur Brown Company (3404 Pawnee Dr., Alexandria, MN 56308) for $19.95 + $1.95 for postage. This program written by Kurt Casby is the most sophisticated BBS for the 2068 yet (all machine code). Another program that is based on the original public domain software, "Tiny-board" by Randy and Lucy Gordon, is also available and used as the BBS for the Indiana Sinclair Timex Users Group BBS.

Casboard and "Tiny-Board" BBS are cropping up all over the place. In the final chapter of Pete Fischer's guide, he predicts that these new boards will have "a major impact" on TS telecommunications.

SINCLAIR QL

The steady stream of business programs, graphic design and CAD programs, utilities (including front-end type software), and game programs, is keeping the QL "alive and well", both in Europe and on a small scale here in the states. There is enough action (and advertising) to keep a genuine "slick" monthly magazine in business--QL WORLD (79-80 Petty France, London, England SW1H 9ED...subscription rate is 45 Brit. pounds for one year...but several TS dealers have them for a discount).



The hottest thing going for the QL, are the second-generation "clone" machines. The 640k CST THOR is available now, complete with 3.5" floppy drive, and IBM style keyboard and packaging. (For further information, write to: Eidersoft, The Office, Hall Farm, North Ockendon, Upminster, Essex, U.K., RM14 3QH...there is even a trade-in offer for your old QL). The newest clone should be out by the time you read this...the Sandy FUTURA. The Futura incorporates several new enhancements like a real 68000 MPU and Tony Tebby's improved QDOS system. (For details, write to: Sandy U.K., Unit 33, Murdoch Road, Bedford, England MK41 7PQ).

While not every QL owner will be upgrading to the newer machine (considering the £600+ price tag), the real value lies in the increase of the over-all QL user base due to clone sales.

A new area for the QL is Desk-Top Publishing with the release of the new program FRONT PAGE. The program isn't on par with PAGEMAKER for the Apple Mac, but Front Page's distributor, GAP Software, promises updates to be available soon. Front Page may spawn other Desk-Top programs. Several U.S. dealers have Front Page in stock now, including Variety Sales, C.W. Associates, Curry Computer, and RMG Enterprises.

A lump comes to ones throat, and a warm feeling to the heart, to see two American QL programs recieve worthy recognition in the European computer market. They are WAR IN THE EAST by Sharp's Inc., and CONCEPT 3D by Tesseract Software (see July/Aug.'86 and Sept/Oct '86 issues of TDM). Both have been rated favorably in

British magazines, and are being distributed by European dealers. Perhaps other American-made programs and hardware may get some notice.

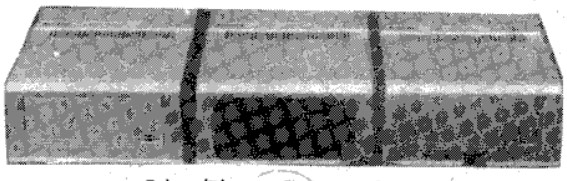
Two soon-to-be-released QL books are coming from the northwest. ARCHIVE MASTER, published by Executive Workshop, will be distributed by RMG Enterprises (1419 1/2 7th Street, Oregon City, OR 97045). The book offers an explanation of Psion's Archive database, and includes several routines and procedures to use the program for serious business applications (inventory, accounts, and mail list management). Archive Master has been "field tested" and comes in a three-ring binder.

Time Designs QL columnist, Mike de Sosa, is the author of TAKING THE QUANTUM LEAP, a 260 page book for both beginners and the advanced QL user. The book will feature much information "missing" from the QL Users Guide, as well as program listings, explanation and enhancements of the bundled software. Over two years in the making! For further information, write to : TDM, 29722 Hult Rd., Colton, OR 97017.

ZEBRA/TIMEX DISK SUPPORT GROUP

First the bad news. Zebra Systems Inc., announced publicly that they have completely sold out of the Zebra FDD Disk Drive System for the TS2068, and that they will no longer carry them.

Now for the good news. A new special interest group is forming to provide support and a newsletter dedicated to users of this disk system. The Zebra FDD is actually a system that is manufactured in a Timex factory in Portugal. These units are marketed in Europe for the Sinclair Spectrum and in Portugal for the TC2068 (a later version of the U.S. TS2068). An early version of the FDD, consisted of three separate components (power supply, controller, and drive). It's disk operating system (TOS) resided in 16k RAM (included), and all three pieces were painted silver like the 2068. A later version, the FDD-3000, contained dual-drives, built-in 64k RAM, CP/M compatibility, all in one compact package, and painted black. Both models use the Hitachi and Amstrad-type 3" diskettes.



Zebra/Timex of Portugal FDD

The new FDD support group's newsletter is called T.O.P.S. (for "Timex Of Portugal Systems") and is for both earlier models of the disk system, and the current FDD-3000. The newsletter's editor is Dave Franson of Milwaukee, Wisconsin. Articles will be written by Dave, Kurt Casby, and many others. Subjects will cover the CP/M operating system, software conversions to disk, use of the RS232 ports, and other related topics. The group is also open to any article or program contributions, and any comments or suggestions.

If you are a Zebra/Timex FDD user, you can contact the group's secretary, John Bylander (1203 N. Owens St. #308, Stillwater, MN 55082) for more details.

Note: The Aerco FD-68 Disc Drive System for the TS2068 has a support group and newsletter also. Users of the FD-68 can contact: Dave Hill, PO Box 310-A, Holland, MI 49423, for further information.

More...

Here's a Whitman's Sampler of some more TS stuff to look for in 1987:

* NOVA 1000 is a new program from Weymil Corp. (PO Box 5904, Bellingham, WA 98227), which performs Multi-Tasking on a TS1000! You can operate multiple programs simultaneously, in addition the programs features a real-time (on screen) clock, auto-repeat for all keys, and program line tracing. All for $20 (plus $2 postage).

* Several GEOS type programs are currently being developed for the TS2068 (GEOS, the front end, housekeeping utility, is currently the best-selling program for Commodore computers).

* An extensive project of de-bugging the TS2068 ROM will be completed soon by Bob Orrfelt of Redwood City, California. A new EPROM will be available to replace the old ROM, and will feature some new routines, including auto line-renumbering.

* A MIDI interface (the standard for electronic musical instruments) is being developed for the TS2068 by Larken Electronics (RR#2 Navan, Ontario, Canada K4B-1H9).

We feel that 1987 will be a good year for Timex Sinclair users. Some of our TS developers also have and use other PC brands. Many of the trends and new products in the computer industry will continue to be adapted (and translated) by these developers for our own microcomputers. Thanks to Uncle Clive, he engineered a very flexible design that can grow and expand for years to come.

Speaking of Sir Clive, he also has some new things up his sleeve to watch for...including Mega-RAM on wafer and the PANDORA portable computer. The beat goes on.

***** SMART TEXT TS-2068 *****

*** A complete Administrative package: Word processor, Data base manager, Mail List, Full text editor, Automatic printer, Automatic formatter. Perfect letters or manuscripts in multiple copies. Mail List for personal form letters. Repeat print 5 page reports. Desk top publishing. Friendly, Menu Driven, no command keys to learn, Crash Proof.
*** Use with ALL Parallel Interfaces, Printers. Second program included for use with OS-64 Cart (64 Col Screen).
*** Disk Versions for Aerco FD and Oliger V2-12.
**** Satisfaction Guarantee. $39.95 Check or M/O, Gulf Micro, 1317 Stratford Ave, Panama City, FL 32404. Specify type interface, for Cassette or Disk Drive, Inquiries welcome.

FRONT PAGE

FOR THE QL BY GAP SOFTWARE

DESK-TOP PUBLISHING FOR THE SINCLAIR QL

AVAILABLE NOW!

From VARIETY SALES

This page set-up using FRONT PAGE

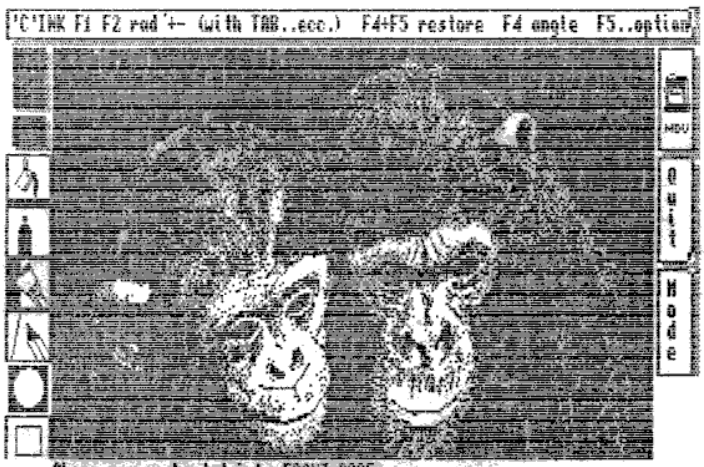
ONLY... $34.95 PP

GAP SOFTWARE ANNOUNCE DESKTOP PUBLISHING ON THE SINCLAIR QL

AVAILABLE EARLY OCTOBER (1986!) RUNS ON UNEXPANDED QL, & SUPPORTS FLOPPIES, RAMDISCS, ETC. Text A N D graphics so you can do your own newsletters, etc. ONLY $34.95 PP!!

Some of the features:

* 16 Typefaces and Sizes Available!
* 64 User Graphics Included!
* Mix Text & Graphics ON SCREEN easily!
* Freehand Drawing Mode!
* Save/Load Screens or Parts of Screens! You can even add screens from OTHER programs into your page! If you have a digitizer, you can add photos!
* Up to 133 columns/88 lines on a page (prints on full 8 1/2" x 11" paper)
* Hardware Requirements: Standard QL with 128K RAM EPSON compatible printer with DOUBLE DENSITY graphics to print full width.. if you have only single density you can print only the first 480 pixels.
* NO EXTRA MEMORY NEEDED! Will work with extra RAM, though. Can also use DISKS or even RAMDISKS! Can easily be transferred to disk! (Disk GREATLY speeds up the program!)
* Whole Program is Menu Driven!
* Ask for Sample Sheet with More Info.



Above screen loaded into FRONT-PAGE after being created with QL PEINTRE!! You can do it too...EASILY!

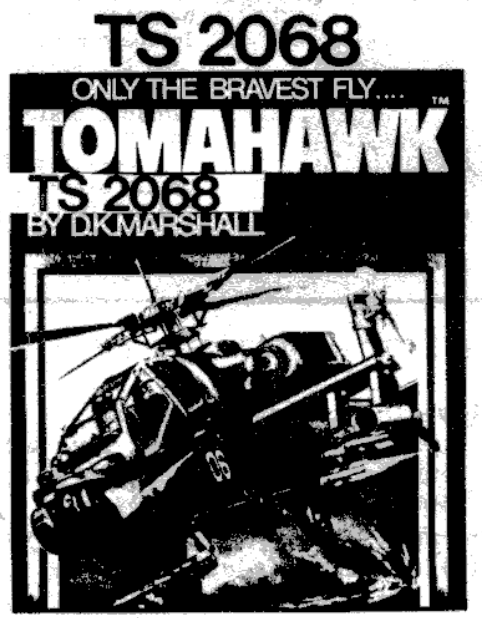
SEND ORDERS TO: VARIETY SALES
325 W. JERSEY ST., #2D
ELIZABETH, NJ 07202 TEL: 201-527-0535

TYPING, PUBLISHING, ARTWORK, ALL AT REASONABLE RATES. ALSO PRINTER RIBBONS FOR SALE. WANT A BOOK PUT TOGETHER? BUSINESS CARDS? RING AND ASK FOR DETAILS. 01-552-5452

TOMAHAWK

TOMAHAWK is a real-time flight simulation based upon the US ARMY AH-64A APACHE Advanced Atack Helicopter - the meanest, deadliest combat helicopter ever to rule the skies! Its specialized job is to hunt tanks and destroy anything that gets in its way. The Apache was built specifically to fight and survive, night and day, in the thick of the battlefield.
Flying a real helicopter is a demanding task, requiring training and practice - particularly ground attack. TOMAHAWK gives you this challenge. Climb into your cockpit and prepare for take-off....

$15.99



FEATURES: Spectacular 3D real world display *Fully aerobatic (within limits of the real helicopter) *Ground attack & air-to-air interception *Over 7000 ground features *Day/Night vision systems *Cloudy conditions, crosswinds & turbulence *Doppler navigation & target tracking system *Laser guided missiles, plus rockets & 30mm chain gun *Selection of training and combat missions *Impressive sound effects *Pilot ratings - Trainee to Ace *Uses BOTH TS2068 joystick ports!

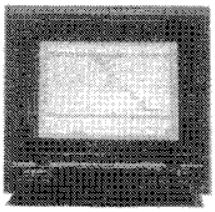
ONLY $15.99!!

Send check or money order to:
Variety Sales
325 W. Jersey St., #2D
Elizabeth, NJ 07202
PHONE: 201-527-0535

(Please add $3 shipping...Thank You!)

MORE "COLORS" IN QL'S MODE 4

By Paul Bingham



Anyone who has a QL and a monitor that will display the mode 4 (F1) screen might be wishing that there was more than four colors; i.e., white black, green and red. If one experiments with the PAPER command he finds that it can provide a stipple function for "blending" colors. Alternating red and white pixels yields a textured pink screen, green and red gives brown and so on.

This short program lets the user pick one of the four pixel patern alternatives then displays labeled overlapping windows of sixteen new "colors". An input window in the lower left allows choices for testing any window or changing the pattern. A small window in the upper right lists the colors and number values and a larger window tests the chosen "color" by printing over it with letters of different sizes and colors. This will reveal that some new pixel-produced "colors" are not suited to certain types of print, while others look just fine.

Refering to the User's Manual while using the COLOR EDITOR will be of value. By altering the window parameters in lines 10, 40, 80, 100 and 200, they offset values in line 50 (and a few other minor changes)...this could also serve as a color editor for mode 8.

```
5 REMark       MODE 4 COLOR EDITOR
10 INK 7:PAPER 0:a=2:z=2:u=2:i=0:WINDOW 512,256,0,0
20 MODE 4:CLS:CSIZE 0,0:AT 0,25
25 INPUT "STIPPLE 0,1,2,or 3? ";s:INK 0:CLS:b=0:f=32
30 FOR d=0 TO 6 STEP 2
35 FOR c=0 TO 6 STEP 2
40 WINDOW 118,60,f,b:PAPER c,d,s:CLS:PRINT" ";
45 PAPER 7:PRINT " PAPER ";c;",";d;",";s;" "
50 b=b+9:f=f+21:END FOR c:END FOR d
60 AT 5,5:PRINT "Stipple: ";s:CSIZE 2,0
70 AT 2,2:PRINT "MODE 4":AT 3,1:PRINT "COLORS'"
80 WINDOW 77,32,265,0:PAPER 6:CLS
85 CSIZE 0,0:FOR c=0 TO 6 STEP 2
90 PAPER 6:PRINT " ink ";c;": ";
95 PAPER c:PRINT "    ":NEXT c:GO SUB 100:GO TO 155
100 WINDOW 97,87,350,0:PAPER a,z,u:INK 0:CLS:PRINT
105 CSIZE 2,0:PRINT "PA";:INK 6:PRINT "TE";:INK 4
110 PRINT "RN";:INK 2:PRINT "S:":FOR t=0 TO 3
115 y$=" '":IF t=1:y$="..":GO TO 130
120 IF t=2:y$=": ":GO TO 130
125 IF t=3:y$="',"
130 CSIZE 0,0:PRINT:INK i:PRINT " Stipple ";
140 INK 0:PRINT t;"= ";:PAPER 4:PRINT y$:PAPER a,z,u
150 i=i+2:INK i:END FOR t:i=0:RETurn
155 GO SUB 200:CLS:AT 1,1:INPUT "PAPER _";a:AT 1,9
160 INPUT ",_";z:AT 1,12:INPUT ",_";u:CLS
170 AT 1,1:PRINT " PAPER ";a;",";z;",";u:GO SUB 100
175 GO SUB 200:AT 2,2:INPUT "new color? (y/n)";n$
180 IF n$="y" OR n$="":GO TO 155
185 GO SUB 200:AT 2,2:INPUT "new stipple? (y/n)";n$
190 IF n$="y" OR n$="":RUN
195 CLS:AT 2,3:PRINT "program ended.":STOP
200 WINDOW 132,32,30,128:PAPER 6:INK 0:RETurn
```

QL GAS GUIDE

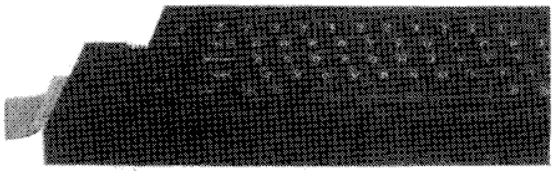
By Gale Henslee

```
10 REMark ******************************************************************
15 REMark ******************************************************************
20 REMark *                                                                *
25 REMark *  PROGRAM TO DETERMINE WHETHER YOU SHOULD SPEND THE EXTRA       *
30 REMark *  MONEY TO PURCHASE PREMIUM GASOLINE INSTEAD OF REGULAR.        *
35 REMark *  IT ASSUMES THAT YOUR CAR RUNS FINE ON EITHER GRADE OF GAS     *
40 REMark *  AND DOES NOT KNOCK OR PING ON THE LOWER GRADE.                *
45 REMark *                                                                *
50 REMark *  GALE HENSLEE          OCTOBER, 1986      AMARILLO, TX         *
55 REMark *                                                                *
60 REMark ******************************************************************
65 REMark ******************************************************************
100 OPEN #1,con
110 WINDOW #1,512,256,0,0:PAPER 6:INK 1
120 BORDER 10,4
130 CLS #1
140 AT 5,5:PRINT "Enter price of REGULAR (d.cc)    ":AT 5,45:INPUT p1
150 AT 6,5:PRINT "Enter price of PREMIUM (d.cc)    ":AT 6,45:INPUT p2
160 AT 8,5:PRINT "Enter miles per gallon you get with REGULAR":AT 8,50:INPUT m1
170 AT 9,5:PRINT "Enter miles per gallon you get with PREMIUM":AT 9,50:INPUT m2
180 d1=p1/m1
190 d2=p2/m2
200 AT 11,5:PRINT "Cost per mile with REGULAR is ";round (d1);" cents."
210 AT 12,5:PRINT "Cost per mile with PREMIUM is ";round (d2);" cents."
220 IF d2<d1 THEN GO TO 250
230 AT 14,5:PRINT "Your best gasoline choice is REGULAR"
240 GO TO 260
250 AT 15,5:PRINT "Your best gasoline choice is PREMIUM"
260 AT 17,5:PRINT "AGAIN? (Y/N)":AT 17,25:LET i$ = INKEY$:IF i$ = "" THEN GO TO
260
270 IF i$="y" THEN GO TO 130:GO TO 330
280 DEFine FuNction round (a)
290 LOCal answer
300 answer = INT (a*1000)/10
310 RETurn answer
320 END DEFine
330 OPEN #1,scr:WINDOW #1,258,200,0,0:PAPER #1,3:INK #1,6:BORDER #1,2,0,7:CLS #1
340 OPEN #2,scr:WINDOW #2,258,200,254,0:PAPER #2,6:INK #2,3:BORDER #2,2,0,7:CLS
#2
350 WINDOW #0,512,57,0,199:PAPER #0,0:INK#0,5:CLS #0
360 STOP
```

Exploring The Mysterious QL

JOS: Adding Your Own Commands to Super BASIC

By Joe Newman



If you're like me and don't know machine language, I'll bet there are times when you really wished you did. You probably would like to make all those fancy application programs with neat advanced features, etcetera. With the power of the QL, you now have an excuse to hold off on learning MC a little longer...Procedures.

Procedures aren't as mysterious as they may at first seem. In fact, they are rather fascinating, and really deserve some attention, even by novice SuperBASIC programmers. Some pretty neat things can be accomplished with them--things that can only be done in machine code on other computers. Here are a few of the things I have discovered about procedures:

1. Procedures typed in as program lines can be called WITHOUT even first typing RUN!

2. Procedures can be called from within other Procedures.

3. Procedure names can be almost anything, INCLUDING SuperBASIC Keywords!

4. Procedures act just like additional commands added to the QL operating system!

You can discover these things yourself by experimenting and reading the QL Users Guide. The BEST way to learn Procedure structure (and anything else on a microcomputer) is by experimenting. You can't blow up the QL by typing in the wrong program, although you can get some pretty strange results!

In this article, I will present a practical use of Procedures involving the above mentioned items. I have created a sample program I call JOS (pronounced JOE's). This program when used will appear to have added and refined some commands to SuperBASIC. To use it, type it in and save it as BOOT. Then put the disk/cart. that contains this program into drive 1 whenever you turn on your QL, so it will load and run automatically. You can even make this the BOOT program on all your disks/carts. If you have I.C.E. you must hold down ALT when you choose monitor (F1) to load the program.

When it first comes on, a screen will say JOS ACTIVE, and a prompt will be waiting for the year. Either continue entering the time and date or hit ENTER at the year prompt. The computer will "jingle" and will instruct you to hit "i", if you wish to go into I.C.E., or b if you want Basic. If you don't have I.C.E., you can delete these program lines.

Once you are in Basic, the only way to tell that JOS is there, is by typing LIST. JOS is from line 20000 on. I have placed it this high so it (hopefully) won't conflict with any other Basic programs. Remember, you can change line numbers with the RENUM command (page 47, Keywords, QLUG). To see a list of all the extra commands available with JOS (so far), just type-in the word "COMMANDS". The following list will be presented:

LD S D DS CAT LPRINT DEFAULT LRUN LLIST CC L SIREN

Here is what each command does and how to use it:

DEFAULT chooses which drive will be the default for the other commands which require use of a drive. Syntax is DEFAULT "name", i.e. DEFAULT "MDV1_". You must add the "_". You can also specify any other type of storage device.

LD will load a program. Syntax is LD "filename".

S will save a program. Syntax is S "filename".

D will delete a file from a storage device. Syntax is D "filename".

DS will delete a file from a storage device, then save the program in memory as that file. Useful for updating programs. Syntax is DS "filename".

CAT will give a directory of the default drive.

LPRINT will print text to the printer through serial port 1. Syntax is LPRINT "text".

LRUN will load and run a program (actually it MERGEs a program with JOS, then runs it). Syntax is LRUN "filename".

LLIST will send a listing of the program in memory to the printer through port 1. Syntax is just LLIST.

CC will clear windows 0,1 and 2; the whole QL default screen. Syntax is just CC.

L will list the program in memory to window #2. Syntax is just L.

SIREN starts a siren sound. To stop it enter BEEP.

To use the commands, either type them in using their proper form, or use them in a new program. I had attempted to incorporate a NEW command, which would delete everything but the JOS program, but I kept getting an error, and the QL would crash. I was trying to use DLINE TO 19000. Can anyone let me know why this won't work, or how I can get it to work? Sometimes it would...most of the time it wouldn't.

Play around, experiment, add your own commands. I have included some of the more useful ones, but I'm sure there are plenty more. Maybe you can add a routine to change the paper, ink, and border, or change windows around. Let me know what you come up with.

By examining this program and figuring out how it works, you will begin to get a good understanding of the power of Procedures. I haven't even started to exploit the full power they have to offer.

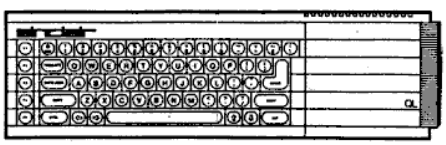
There is one feature of Procedures I would like to explain. While writing JOS I discovered for myself (by accident) that strings could be added after procedure names; I had thought only numbers could be passed into procedures. An example of this is:

```
DEFine PROCedure say (x$)
PRINT x$
END DEFine say
```

The proper name of this procedure is "say". If you had this routine running on the QL, and then typed...say "Hello---it worked!" and enter "hello---it worked! will appear in window #1 (the output window, red). That is how I made the Procedures, such as DEFAULT. DEFAULT is the actual procedure name, and you enter a string after it, which contains other information. You must remember the quotes, though! Also, if JOS crashes with an error, just try again...no harm done.

Remember, experimenting and doing is the key to learning. Fool around, try anything. If it doesn't work, keep trying, or else scrap it and start over! To constantly resave your own altered version of JOS as you work on it, use the DS command. It will allow you to save your current version, while you can continue altering. Remeber to save it as BOOT so it will auto load and run.

Let me know how you came along in your experimentations. If you have any questions or comments, let me know also. Joe Newman, 325 West Jersey Street #2D, Elizabeth, NJ 07202.



JOS

```
20000 cc:CSIZE 2,1:PRINT " *** JOS ACTIVE ***"
20010 PRINT                "@1986  by JOE NEWMAN"
20020 CSIZE 0,0:PRINT "Enter year or ENTER to quit":CSIZE 3,1:INPUT "year (yy) "
,year$:IF CODE(year$)<>0 THEN year=year$: INPUT "month",month:INPUT "day",day:IN
PUT "hour",hr:INPUT "Min.",min:year="19"&year:SDATE year,month,day,hr,min,0
20030 CSIZE 0,0:BEEP 0,1,255,1300,1
20040 drive$="flp1_"
20050 PAUSE 50:BEEP:PRINT  "hit > i)ce or b)asic"
20060 ink$=INKEY$:IF ink$="" THEN GO TO 20060
20070 IF ink$="i" THEN ICE:ELSE cc:STOP
20080 DEFine PROCedure cc:CLS#0:CLS#2:CLS:END DEFine cc
20090 DEFine PROCedure l:LIST:END DEFine l
20100 DEFine PROCedure ld (file$)
20110 call$=file$
20120 file_name$=drive$&call$
20130 MERGE file_name$
20140 END DEFine ld
20150 DEFine PROCedure default (driver$)
20160 drive$=driver$
20170 END DEFine default
20180 DEFine PROCedure LRUN (file$)
20190 call$=file$
20200 file_name$=drive$&call$:MERGE file_name$
20210 RUN
20220 END DEFine LRUN
20230 DEFine PROCedure llist
20240 OPEN #100,ser
20250 cc:PRINT "READY PRINTER (hit any key)":PAUSE 40000
20260 cc
20270 LIST #100
20280 CLOSE #100
20290 DEFine PROCedure cat
20300 DIR drive$
20310 END DEFine cat
20320 DEFine PROCedure s (file$)
20330 call$=file$:SAVE drive$&call$
20340 END DEFine s
20350 DEFine PROCedure lprint (printer$)
20360 OPEN #100,ser:PRINT #100,printer$:CLOSE #100
20370 END DEFine lprint
20380 DEFine PROCedure d (file$)
20390 DELETE drive$&file$
20400 END DEFine d
20410 DEFine PROCedure ds (file$)
20420 call$=file$
20430 DELETE drive$&file$
20440 SAVE drive$&file$
20450 END DEFine ds
20460 DEFine PROCedure commands
20465 cc:PRINT "AVAILABLE COMMANDS FOR JOS"
20466    PRINT "--------------------------"
20470 RESTORE 20480:FOR LOOPER=1 TO 12:READ COM$:PRINT COM$:NEXT LOOPER
20480 DATA "LD","S","D","DS","CAT","LPRINT","DEFAULT","LRUN","LLIST","CC","L","S
IREN"
20490 END DEFine commands
20500 DEFine PROCedure siren:BEEP 0,1,1300,150,1:END DEFine siren
```

QL Quill/Word Processor Tips

PART II

By Mike de Sosa

I'd like to start off this article with a few tips regarding the QL QUILL commands. I assume that you are already familiar with the use of these commands.

The Copy command is not user-friendly. Follow the associated screen instructions carefully or you will, almost inevitably, lose desired material when copying. If large segments are to be copied, Save the document prior to doing so. The Copy command offers you two options in a rather tricky sequence: you can copy a passage from one location to one or more other locations, first deleting it from its former position, or you can copy it without first deleting it. Simple enough, but watch your step!

The Design command sequence can be used to ease your eyes. In most cases, your QUILL document formats should be no more than 64-columns wide. Set the "Display width" to 64-columns for easier viewing; you may then, like me, find that you no longer need your spectacles to use QUILL. If you still experience eyestrain, try alternating between green and white letters for your main text.

In using the Erase command, keying SHIFT and the Down cursor highlights text to be erased a paragraph at a time. Keying SHIFT and the Right cursor highlights text a word at a time. If you go too far, use the Left and Up cursor (with or without the SHIFT key) to back up.

Make full use of the Header and Footer commands. Use Header to make single-line letterheads (use a specially prepared letterhead "_doc" file for multi-line letterheads). Use Footer to include your address and telephone number, ect., in a single-line continuation of your letterhead at the bottom of each page. Don't forget that there is a "default" Footer that must be canceled, if not desired.

Don't forget to use Goto to move through your document. This saves both time and wear and tear on your QL keyboard.

Justify can be used as a time-saver when entering

FREE MONITOR! and I.C.E.

YES! IT'S TRUE!! Just order a QL Computer and a QL printer from VARIETY SALES and you will get, absolutely FREE, a Magnavox (or equivalent brand name) Monochrome Monitor along with the proper cable and everything else you need to start a small business or home computer set-up:

A+ Computer Response Authorized Dealer

===> Sinclair QL with 90 day warranty
===> Magnavox (or equivalent) monochrome monitor
===> Sinclair QL Near Letter Quality Printer
===> Psion Quill full-feature Word Processor
===> Psion Archive full-feature Data Base
===> Psion Easel Business Graphing/Charting
===> Psion Abacus Electronic Spreadsheet
===> All cables, manuals, wallet of 4 microdrive cartridges, power supply, ribbon for printer

And you get it all for just: $599

Send check or Money Order for $599 +$12 shipping to:

VARIETY SALES
325 WEST JERSEY STREET #2D
ELIZABETH, NJ 07202
PHONE: 201-527-0535

The WIDJUP Co. presents

TOURIST C

BANK SWITCHING IS HERE! BE READY FOR IT.

Tourist C is really an extended bank switching disassembler and SPY program residing in BASIC. It uses machine code located above "COPYUP" in the machine stack. Printing to the 2040 is not usually desirable, so a universal interface is included. When used, the appropriate kernel is loaded into the printer buffer. Because this is an "overlay" it does not interfere with any usage by other banks or peripherals.

To help convince you of the great features of this program, send us no more than 60 bytes of any code you like and SASE. The WIDJUP Co. will return a disassembly of that code and more info about TOURIST C. How's that for bait? Try it.

The WIDJUP Co.
1120 Merrifield S.E.
Grand Rapids, MI 49507

Program: TOURIST C
Order #: T525PY86B
Price: $32.50 inc P&H

WINKJET 1 lets you use All the features of your OLIVETTI PR2300 ink jet printer.

PIO#2 is a MENU built universal interface. Use your TASMAN, AERCO, or home brew parallel physical interface.

LPRINT speaks fluent extended ASCII, and is adept at the PR2300 GRAPHICS dialect.

WCOPY dumps the screen to the printer in normal size or ZOOM. LLIST is supported in high resolution graphics using WCOPY.

WHAT YOU SEE IS WHAT YOU GET!

WINDOW dumps part of the screen in variable length lines up to 110 characters per line.

Its default configuration prints the lower two screen lines as 64 wide.

The WIDJUP Co. word processer/data base program TypaLot uses WINDOW to prepare ads like this and the one to the left.

Program: WINKJET 1
For: TS2068 w/OLIVETTI PR2300 printer
Order #: T70INTF86B
Price: $14.95 plus $1.50 S&H

WIDJUPEEL WINDOWS on GORILLA BANANA...Order#T71INTF86B $14.95 + $1.50 S&H
TypaLot J1 WORD PROCESSING on PR2300...Order#T51WD5J86B $18.50 + $1.50 S&H
TypaLot GB Ditto for GORILLA BANANA....Order#T51WD5B86B $18.50 + $1.50 S&H

multi-line, centered headings or titles. Executive-level correspondence is not usually right-justified; use proportional spacing or "near letter quality" (High Quality on your QL printer) typeface for this, if your printer has such features. The justification selected applies to the current paragraph and all following text, until you again change justification.

Prepare short "_doc" files for each of your frequently used document formats so you don't have to set-up and format QUILL each time you wish to prepare such a document. Include such things as letterheads; Design settings, margin, and tab settings; Headers and Footers; introductory paragraphs, closures, and signature blocks. SAVE these to your program cartridge or disk and Load the appropriate "_doc" file when beginning a new document. Include the title and purpose of this mini-document and its parameters in text. Don't forget to change the name of your document file before SAVEing it for the first time or you will overwrite it.

Use Margins command efficiently and imaginatively. If you set the Indent margin to the same value as the Left margin and use ENTER as you would a carriage return, you are misusing QUILL. ENTER (or the Down cursor key) should only be used to begin a new paragraph; otherwise, scrolling by paragraph for any purpose will be greatly slowed. Instead, use the SPACE bar to move the cursor to the end of a line, or, better yet, the TABULATE key: neither of these methods initiate a new paragraph. Remeber, too, that the Indent margin may be set to the left of the Left margin, useful for typing numbered paragraphs and for other purposes.

Remember that the Print command may be used to print a hard copy of all or a portion of any QUILL ("_doc" suffix) document on any on-line medium--a very useful feature, if you keep a good index of just what is on what page of certain key documents. Note that in the Print command sequence you may call for a directory of the data medium in the usual manner. Try this, you'll like it.

Print may also be used to print all or part of any QUILL document to a Microdrive or other type file. Such a file would have the "_lis" suffix (unless another suffix, e.g., "_exp" were specified) and cannot be Merged or Imported into QUILL, not printed using the QUILL Print sequence. Such files are useful for a number of purposes, including "print spooling", i.e., printing such documents without QUILL or alongside QUILL using special software, or printing such documents from SuperBASIC using:

COPY MDV2_QUILL_lis TO SER1

The "_lis" and "_exp" files produced using the Print command are printed or imported in print fashion, that is, if they are double-spaced files, they would be printed double-spaced on your printer or on your monitor screen.

The print-to-a-file option of the Print command sequence may be used to print an export file (the "_exp" suffix must be appended to the filename) from any document file on storage media. This file may then be imported into the current QUILL document file using the Import option of the Files command sequence. This is a quick and excellent method of "cutting and pasting" passages from among several QUILL documents, expecially if the working files are loaded onto RAMdisk.

Finally, before starting to print, QUILL reads the current printer driver (printer_dat file) information from your designated (using config_bas) program medium (i.e., MDV1_, FLP_, RAM1, ect.). This printer driver data may be selected from among several sets of such data from within QUILL using the Delete and Backup options of the Files command sequence. For example, store several printer drivers on your program medium as printer1_dat, printer2_dat, ect. Then, before printing, decide which printer driver is to be used and delete the current printer_dat file using the Files command option:

Delete mdv1_printer_dat

Then copy the appropriate printer driver to printer_dat using the Files command option:

Backup mdv1_printer2_dat to mdv1_printer_dat

You are now ready to print, and your selection of printer drivers is intact.

It may seem to you that the Quit command does not always work in the same fashion. If the current QUILL document is amended in any way, you must either Save the document or take a positive measure to zap it before you are permitted to Quit, otherwise, you may quit almost directly.

Make full use of Tabs, but do not use more than you need as this will slow down using the TABULATE key as your carriage return as recommended above.

The Files command options (Backup, Delete, Format, and Import) assume that (default) your designated data medium is the device desired unless this is otherwise specified. Backup and Delete assume "_doc" file suffixes unless otherwise specified. Import will only import files with the "_exp" suffix. Frequently overlooked, these command options offer good flexibility in file management, cutting and pasting documents, and selecting alternative printer drivers.

Use of the Hyphenate command is good practice, but incredibly slow. If you have some kind of "key define" software, this is a good candidate for translation to one or two keystrokes. Use Hyphenate during final editing of a manuscript to reduce overly wide gaps on the printed line: just remain in the Hyphenate sequence as you scroll through the entire document.

For safety's sake, always Save the current document before attempting to Merge it with another. If only a minor part of another document is needed, consider use of the QUILL-to-QUILL export method described above.

Like Hyphenate, the Page command is a good candidate for translation to a single or double-keystroke. I have found that QUILL ofter "locks up" when I attempt to eliminate a page break and recommend that you Save the current document to a permanent storage medium before attempting this.

Use the Replace command to semi-automatically search for and correct words that you frequently misspell or to correct a word that you later find was misspelled. Replace may also be used for many other purposes, for example, completely deleting certain words or punctuation marks from your document.

DRIVER NAME	:	QL PRINTER1	
PORT	:	SER1	
BAUD RATE	:	9600	
PARITY	:	NONE	
LINES/PAGE	:	66	
CHARACTERS/LINE	:	80	
CONTINUOUS FORMS	:	YES (if using fanfold paper) or NO (if not)	
END OF LINE CODE	:	CR,LF	
PREAMBLE CODE	:	27,64,27,82,0	ENTER
POSTAMBLE CODE	:	NONE	
EMPHASIZE ON	:	27,69	ENTER
EMPHASIZE OFF	:	27,70	ENTER
UNDERLINE ON	:	27,45,1	ENTER
UNDERLINE OFF	:	27,45,0	ENTER
SUBSCRIPT ON	:	27,83,1	ENTER
SUBSCRIPT OFF	:	27,84	ENTER
SUPERSCRIPT ON	:	27,83,0	ENTER
SUPERSCRIPT OFF	:	27,84	ENTER
TRANSLATE1	:	96,27,82,3,35,27,82,0	ENTER
TRANSLATE2	:	"CS/P,27,112,1,14,27,71,"SPACE	ENTER
TRANSLATE3	:	"CS/I,27,52,27,69,27,71,"SPACE	ENTER
TRANSLATE4	:	"CS/S,27,72,27,70,27,53,"SPACE	ENTER
TRANSLATE5	:	"CS/E,27,77,"SPACE	ENTER
TRANSLATE6	:	"CS/R,15,27,71,"SPACE	ENTER
TRANSLATE7	:	"CS/B,27,71,"SPACE	ENTER
TRANSLATE8	:	"CS/H,27,120,1,"SPACE	ENTER
TRANSLATE9	:	"CS/Q,27,120,0,"SPACE	ENTER
TRANSLATE10	:	"CS/Z,27,112,0,18,27,80,27,72,"SPACE	ENTER

Figure 1. QL Printer Typeface Codes

QUILL Typeface Options

There are several ways to insert additional typeface commands in a QUILL document. Most of these require use of the INSTALL_BAS (see the "INFORMATION" section of the QL User Guide) program to modify the preamble code, the standard QUILL typeface codes, or the ten "translate" options offered on each printer driver.

Printing is controlled by a special program called the printer driver or PRINTER_DAT file which may be modified to work with different printers and to produce customized results. Appropriate printer driver data for a number of printers and your customized data is stored in the INSTALL_DAT file. The INSTALL_BAS program is used to select and modify (customize) printer driver data and to "install" your selection in the current PRINTER_DAT file. QUILL, itself, uses only the PRINTER_DAT file on the program medium which it consults before printing each document.

The examples given below are for the Sinclair QL Printer. Use your printer manual to modify the examples shown for your printer set-up.

Figure 1 is a depiction of the printer driver information in the edit mode of INSTALL_BAS. In this depiction, decimal codes are shown for the entries under PREAMBLE CODE, ect. Select the FX-80 line and key F1 to make a copy of it, and then F2 to edit it to be like Figure 1. When you have entered the decimal code for a given line, check it very carefully and then ENTER; at this time, the decimal codes will change to alphanumeric codes. To set in the codes for TRANSLATE2, et seq., proceed as follows:

a. Select the line for editing with the right cursor key.

b. Type a " (using SHIFT '), then CONTROL SHIFT P followed by a comma, then the following numbers each followed by a comma: 27,112,1,14,27,71,

c. Following the comma after the last number, type ", a space (using the space bar), and then key ENTER.

In explanation, "CS/P and the symbol it produces signal QUILL to send decimal codes to your QL Printer: 27,112,1 are the code for proportional spacing; 14 the code for double-width printing; and 27,71 the code for double-strike (darker) printing. In a QUILL document then, keying CONTROL SHIFT P depicts the Greek letter mu on the screen and orders the QL Printer (and many other printers) to print the following text in double-width, double-struck letters with proportional spacing.

CS/I directs bold, double-struck italics.
CS/S terminates italic printing.
CS/E directs Elite (12 cpi) printing.
CS/R directs double-struck Condensed (17 cpi)print.
CS/B directs bold, double-struck print.
CS/H directs High Quality printing (use the decimal codes 27,66,4 with other printers--or check your own printer manual).
CS/Q terminates High Quality printing (use the decimal codes 27,66,5 with other printers--or check your printer manual).
CS/Z terminates the following:

1) Proportional spacing
2) Elite (12 cpi) printing
3) Condensed (17 cpi) printing
4) Double-strike printing

Experiment using all of these in a text document. Become familiar with the symbols and effects each command produces. Check which may be used in combination (there is a priority system which precludes some). Check your printer manual to see what else may be done, e.g., Mode Combinations using the QL printer. Most printers should make use of these codes with only minor changes. If you have any questions, write to me c/o TIME DESIGNS.

A Promising New Product

QATS, a powerful QL utility designed to reduce the number of keystrokes required to perform housekeeping chores and manage other computer tasks, is pronounced "cats" because it gets rid of "mice". QATS, for QL Applications Traffic Supervisor, is, among other things, an attractive alternative to such programs as I.C.E., with or without a mouse (though not for CHOICE, since it does not--as yet--have a multitasking capability).

QATS is "menu-driven" (menus may be tailored to your needs)--only two keystrokes are required to call any of the Psion programs--and provides facilities for the following tasks:

a. Initiating and returning from the Psion programs.
b. Initiating, managing, and returning from other EXEC'able programs.
c. Sophisticated wildcard filename processing.
d. Fast file copying and deletion--en masse or selectively.
e. Formatting media.
f. Alphabetical media directories with file lengths
g. Comprehensive printing package, including print drivers and spooling.

Due to an equipment malfunction, I cannot give a fuller description of how QATS works at this time, but I did want to call this very promising piece of software to your attention. QATS is available from: COPE, 3 Langham Mansions, Earls Court Square, London, England, SW5 9UH, U.K. QATS and its Output Control (an extension of QATS) sell for $35 and $25, respectively and are available in and EPROM version.

That's all for now. Next time: more on QL QUILL, including using it with floppy disks, RAMdisk, and some exotic software.

Now Available! "THE BEST OF SUM, PART II"

Over 60 pages of program listings, reviews, hardware projects, hints and tips, and articles. Covers TS-1000, TS-2068, and QL. All are reprints from the last year of **SUM** plus a few that didn't make it into print before now.

Articles include Building an EPROM Programmer, Sprites on the 2068, Adding RGB to 2068, QL Word Processing, What's Available for TS-1000, and much more. Does overlap the "Special Issues" advertised to the right.

$6.95 postpaid

The original **"THE BEST OF SUM"** is now in its 2nd printing. 112 pages of articles going all the way back to the beginning of **SUM** — 3 years worth!

$11.95 postpaid

QL Special Issue (12/85) — $3.95 postpaid.

TS-1000 Special Issue (5/86) — $3.95 postpaid.

Everything! $20.00 postpaid

FOOTE SOFTWARE P. O. Box 14655 — Gainesville, FL 32604 904/462-1086 (6 pm - 9 pm EDT)

2068 TAX CALCULATOR

By Herb Bowers, Sr.

One of the most rewarding and interesting aspects of programming is in designing accurate and meaningful programs for business application...and our TS2068 (and TS1000) are highly adaptable to this purpose. (Mr. Sugar please pay attention!)

Being a retired Federal Auditor, I have always had a profound interest in accounting and tax programs. I currently keep several sets of books for small business in my area and rely entirely on my 2068 to do most of the work. During tax season for the past two years, I have used my 2068 (and my 1000 before that), to perform the complex calculations required in the preparation of my clients returns.

I have seen several tax programs that have been produced during the past several years. I have even had occaison to work with one highly praised program written for the Apple. For the most part, the programs (including the one for the Apple) left much to be desired. They appeared to be "word processor" programs with an add/subtract function...you still needed a calculator, a scratch pad, and a stack of charts and schedules and a lot of patience to operate them.

A truly dedicated income tax program should require you to have nothing on your desk but your computer, your W2's, some receipts and a cup of coffee. It may be helpful to have a manual nearby (Lasser is about the best), just in case you have to check some theory, which is impossible for your computer to do (so far).

The main program itself is a snap to write. After all, the entire pseudo-code is already written for you on the tax form itself. The only thing you need to do is assign variables to the lines on the form, and you're in business. Setting parameters for some items would be helpful and this would allow the computer to alarm you for values that could trigger an audit. These figures can be found in most good manuals (but not from the IRS), and are programmable.

That brings us to the various calculations required to determine the taxable or deductible values of certain items on the return (i.e., Social Security, retirement income, IRA's, marriage deduction, ect.). These can be found in the instructions for the official forms, and are easily programmable as sub-routines.

All of the attached schedules can be easily programmed including schedule "C" and simple schedule "D". Schedule "G" presents quite a challenge for the programmer, but can be done. This one is really fun to write and you have a really warm feeling of accomplishment when you complete it.

This brings us to the actual tax calculation itself. Equations used to calculate taxes as with most accounting equations require only basic arithmetic skill (addition, subtraction, multiplication and sometimes division). In fact if any higher mathematics were essential, I would have gone into worm-farming instead of accounting!

Now two methods are used to determine tax liabilities. Note that I said "are" used, and not "may" be used. There are the X, 2Y's(?), and Z Tax Rate Schedules and then there are the ever popular Tax Tables.

The Internal Revenue Service in their infinite wisdom, with the help of Congress, gives us no choice. In fact, that freedom is not even addressed in the U.S. Constitution. If your taxable income is less than $50,000 per tax year you MUST use the TAX TABLES to calculate your tax liability. (An exception is the use of schedule "G", and the use of form 4972--Special 10 Year Averaging, which only determines the tax on specific types of income and not your total taxable income).

Now looking up a figure on a Tax Table appears to be a fairly simple thing to do, and one would think that any eleven year old would accomplish it with ease. But not only are the Tax Tables (with their 4,932 separate brackets) NOT easy to use, but they are one of the great inequities that exist in the current tax system. Actual IRS statistics show that more HONEST mistakes are made in this area than any other.

Tax Reform is not the subject of this article, so now let's get back to business. The IRS in preparing the Tables did not just pull the figures out of the air, but did apply a specific formula. You can't call the local IRS office to get the formula, because I doubt that the "voice" on the other end of the phone would even know what it is.

It took a few hours of mostly "hit and miss" calculating, but I finally figured out the precise formula used by the IRS programmers. Once I had the equation, breaking down the XYZ schedules into DATA statements was no problem. Then I set out to write the algorythm to calculate the tax. It is necessary to calculate the tax twice at both the upper and the lower bracket parameters for the particular taxable income. This should produce the exact tax found in the Tax Table. Simple, right? Wrong!!! For some reason the IRS computers, billed as the "most sophisticated computer system in the world", goof-up once in awhile. This is not the ordinary computer goof that may occur due to binary conversions, but goofs in entire sections of the tables. This year, it only occured in one small section of the tables (which is an improvement over past years). I told an auditor at a local IRS office about it last year. He just smiled and said, "Well it's only a dollar off!". True, but it's our dollar.

Since most of us MUST use the Tax Tables, we must also use the correct figures. Therefore, in the Tax Calculator program that follows, some calculations within the DATA itself were necessary so that our 2068 could compensate for the inadequateness of the IRS computers. Line 9512 [DATA 131.9-sch] Everyone with taxable income from $3,700 to $5999.99 will be charged an extra buck. Line 9522 [DATA 0-((ta>3670 AND (ta>3700))+sch)] Taxpayers with taxable amounts $3670 to $3688.99 will get a $1 break. Hey IRS! "It's only a dollar!".

All that is left for you to do now is to type-in the sub-program and work it into your own main program. The program itself is lines 1100 thru 9544. The INPUT and OUTPUT variables are detailed in the REM at 1100.

Form **1040** Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return

For the year January 1-December 31, 1984, or other tax year beginning , 1984, ending

Use IRS label. Otherwise, please print or type.	Your first name and initial (if joint return, also give spouse's name and initial)	Last name
	Barry R. & FAYE	DOUGH
	Present home address (Number and street, including apartment number, or rural route)	
	1000 CITY DRIVE	
	City, town or post office, State, and ZIP code	Your occupation
	ANYPLACE USA 07942	Spouse's occupation

2068 TAX CALCULATOR

Lines 100 thru 160 are a test module. Plug in some figures and then check them against the Tax Tables that came with your 1040. Since the test module is only intended to test Form 1040, 1040A & EZ, it is necessary to put in a dummy variable (sch=0) so that the program will calculate Tax Table tax. If you are going to calculate tax for other forms or schedules, "sch" should be set to 1 (true).

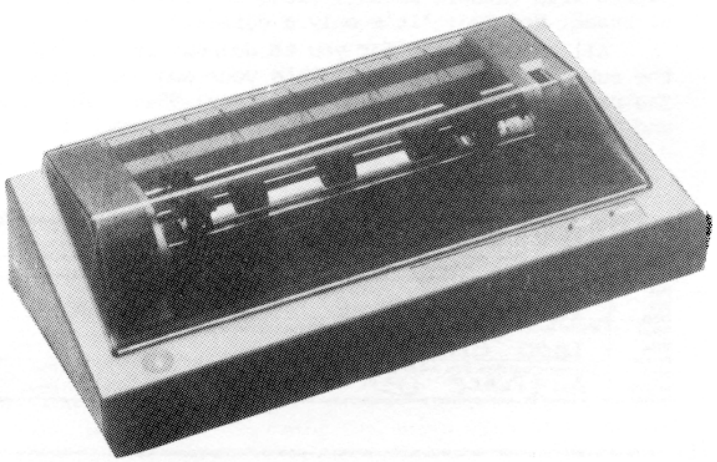
I would like to get some "feedback" as to how you liked this program, and if you would like to see some more along the same lines. You can either write to TIME DESIGNS, or you can write to me direct. I'll give you "sudden service" on answering any questions. You can locate me here, behind the same old stand at 2588 Woodshire Circle, Chesapeake, VA 23323.

```
  0>REM         1986
    **Federal Tax Calculator**
       by: herb bowers, sr.
           December 1986

 100 REM ■Test Module 1■
 110 LET sch=0:
     REM dummy variable

 120 INPUT "Filing status (1 TO
5)?";fs: PRINT "Filing Status: "
;fs
 130 INPUT      ("Taxable Income:
"" 1040, Line 37"" 1040A, Line
 19"" 1040EZ, Line 7");ta: PRIN
T "Taxable Income: $";ta
 140 PRINT "Tax Liability: $";
 150 GO SUB 1100: PRINT tax''
 160 GO TO 120

1100 REM ■tax calculation■

     INPUT: ta=taxable amount
            fs=filing status
               (1 to 5)
          sch=Any schedule or
              Form other than
              1040, 1040A or EZ

     OUTPUT: tax=tax liability

1110 LET l=ta: LET l=l/100:
     LET data=9500+(fs*10)
     -((fs=5)*30)
1120 IF NOT sch AND ta<50000
     THEN GO TO 1140
1130 GO SUB 1200:
     LET tax=INT (l*100+.5)/100:
     RETURN

1140 IF ((fs=1 OR fs=4) AND
     ta<=2480) OR ((fs=2 OR
     fs=5) AND ta<=3670) OR
     ((fs=3) AND ta<=1850) THEN
      LET tax=0: RETURN

1150 LET br=25+(((l>30)*25):
     LET b1=INT (ta/br):
     LET b2=b1*br: LET b3=b2+br:
     LET l=b2/100

1160 GO SUB 1200: LET l1=l:
     LET l=b3/100

1170 GO SUB 1200: LET l2=l

1180 LET tax=INT ((l1+l2)/2+.5)
1190 RETURN

1200 RESTORE data: LET lo=0

1210 FOR f=1 TO 15+(fs=1)
1220 READ hi,plus,pct
1230 IF l>lo AND l<=hi THEN
       LET l=(l-lo)*pct+plus: RE
TURN
1240 LET lo=hi: NEXT f: RETURN

9500 REM ■tax calculation data■

9510       REM ■single■
9512 DATA 24.8,0,0,36.7,0,11,47.
5,131.9-sch,12,70.1,260.5,14,91.
7,576.9,15,116.5,900.9,16,139.2,
1297.7,18,161.9,1706.3,20,196.4,
2160.3,23

9514 DATA 253.6,2953.8,26,310.8,
4441,30,368,6157,34,447.8,8101.8
,38,596.7,11134.2,42,882.7,17388
,48,1E36,31116,50

9520      REM ■joint & q/w■
9522 DATA 36.7,0,0,59.4,0-((ta>3
670) AND (ta<3700))+sch,11,82,24
9.7,12,128.4,520.9,14,172.7,1170
.5,16,218,1879.3,18,265.5,2694.7
,22,322.7,3739.7,25

9524 DATA 379.8,5169.7,28,494.2,
6768.5,33,647.5,10543.7,38,923.7
,16369.1,42,1180.5,27969.5,45,17
52.5,39525.5,49,1E36,67553.5,50

9530 REM ■married filing sep.■

9532 DATA 18.35,0,0,29.7,0,11,41
,124.85,12,64.2,260.45,14,86.35,
585.25,16,109,939.65,18,132.75,1
347.35,22,161.35,1869.85,25

9534 DATA 189.9,2584.85,28,247.1
,3384.25,33,323.75,5271.85,38,46
1.85,8184.55,42,590.25,13984.75,
45,876.25,19762.75,49,1E36,33776
.75,50

9540  REM ■head of household■

9542 DATA 24.8,0,0,47.5,0,11,70.
1,249.7,12,93.9,520.9,14,127.3,8
54.1,17,161.9,1421.9,18,196.4,20
44.7,20,253.6,2734.7,24

9544 DATA 310.8,4107.5,28,368,57
09.1,32,482.4,7539.5,35,653.9,11
543.5,42,882.7,18746.5,45,1168.7
,29042.5,48,1E36,42770.5,50
```

GRAPHEX-PANDER

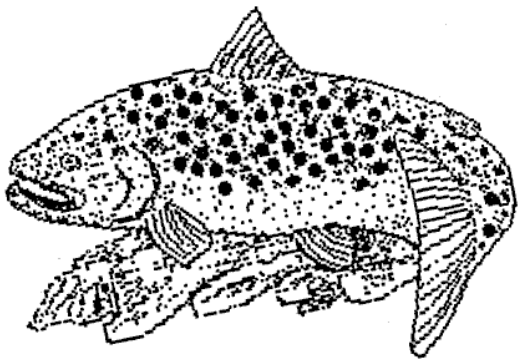
Sideways Graphics Utility for Gorilla Banana

By John McMichael



The following program is a BASIC screen dump program for the Gorilla Banana printer with the Aerco (or Oliger) printer interface. The "heart" of the program is the POINT (x,y) command. What it does is to test screen pixel (x,y) and return a 1 if it is INK color or a 0 if it is PAPER color. The program uses this information to expand each INK colored screen pixel into the resulting 2X2 dot printout.

To use the program, first LOAD, MERGE, or type in the GRAPHEX-PANDER program. Next LOAD or otherwise produce the screen that you want printed. Turn the printer ON and type GO TO 9800. DO NOT RUN the program as the screen will be cleared. Be patient...the program is written in BASIC and is therefore painfully slow. A full screen print-out will take approximately 21 minutes.



I have compiled the program and it does a full screen printout in only 4 minutes. It is 576 bytes long and resides at 60000. For those of you wanting this faster machine code version, please send $4.00 to: John McMichael, 1710 Palmer Dr., Laramie, WY 82070. I will also include a copy of the BASIC program as it is presented here, on the same tape.

The sample graphics screen printout shown, was made with the program ART STUDIO and GRAPHEX-PANDER.

GRAPHEX-PANDER

```
9800 REM Put Banana into GRAPHIC MODE & initialize variables.
9805 LET t=8: GO SUB 9950: LET x=0
9810 LET y=0
9815 LET t=128
9820 REM Expand to 2 dots X 2 dots each "ON" screen pixel using POINT (x,y).
9825 IF POINT (x,y) THEN LET t=t+3
9830 LET x=x+1: IF POINT (x,y) THEN LET t=t+12
9835 LET x=x+1: IF POINT (x,y) THEN LET t=t+48
9840 LET x=x+1: IF POINT (x,y) THEN LET t=t+64
9845 REM Loop to build a line of graphics in printers' buffer.
9850 GO SUB 9950: IF y<175 THEN LET y=y+1: LET x=x-3: GO TO 9815
9855 REM Print the graphic line.
9860 LET t=13: GO SUB 9950
9865 LET y=0
9870 REM Expand to 2 dots X 2 dots each "ON" screen pixel using POINT (x,y).
9875 LET t=128: IF POINT (x,y) THEN LET t=t+1
9880 REM If x=255 then GO TO 500 for special last line print-out.
9885 IF 254<x THEN GO TO 9930
9890 LET x=x+1: IF POINT (x,y) THEN LET t=t+6
9895 LET x=x+1: IF POINT (x,y) THEN LET t=t+24
9900 LET x=x+1: IF POINT (x,y) THEN LET t=t+96
9905 REM Loop to build a line of graphics in printers' buffer.
9910 GO SUB 9950: IF y<175 THEN LET y=y+1: LET x=x-3: GO TO 9875
9915 REM Print the graphic line.
9920 LET t=13: GO SUB 9950: LET x=x+1: GO TO 9810
9925 REM Special branch for last line print-out. Stops & returns printer to
     NORMAL MODE when done.
9930 GO SUB 9950: IF 174<y THEN LET t=13: GO SUB 9950: LET t=15: GO SUB 9950:
     STOP
9935 LET y=y+1: IF POINT (255,y) THEN LET t=129: GO TO 9930
9940 LET t=128: GO TO 9930
9945 REM Output to printer via Oliger or Aerco centronics I/F.
9950 LET b=1: IF 16<t THEN LET b=2
9955 IF IN 127=253 THEN GO TO 9955
9960 OUT 127,t: LET b=b-1: IF b THEN GO TO  9955
9965 RETURN
```

T/S MODEM-ing

RUN LENGTH ENCODED GRAPHICS

By Stan Lemke

Even though you might have the very best graphics programs for the TS2068, it still takes a lot of time and talent to create anything that remotely resembles art. Wouldn't it be nice to get copies of computer graphics that..."more talented" people created for their computers?

Well, there is! COMPUSERVE (for example) has libraries of computer graphics that you can "download" to your computer and enjoy...with a minimal amount of effort! What you need is a modem, a subscription to CompuServe (for example), and the following RLE decoder program. RLE graphic files are hi-resolution graphic pictures that follow a "standardized" format compatible with just about any computer supporting high resolution graphics. The TS2068 will support RLE Decoded graphics in the 32 column video mode.

John Ryan has supplied to CompuServe subscribers, several files related to the subject of RLE graphics: an ENCODE program used to take TS2068 graphics files (like those of the Pixel Sketch and Graphics Editor--for example) and convert them to the RLE format to upload to CompuServe. There is a DECODE program to convert an RLE download picture to the TS2068 format; and there is an information file that goes into detail concerning RLE format structures. For now, we will only be concerned with the DECODE program.

The following program is a modification of the one supplied by John Ryan. The program performs thousands of calculations in the conversion process (RLE to TS2068) and is very time-consuming in BASIC. It will work as is, but very, very slowly. For all practical purposes, one really needs to compile it. LISTING #1 presents the program as used directly from BASIC. Simply type in the program, SAVE it to tape by typing: RUN 9999 [ENTER]. When you LOAD the program it will auto-run and begin prompting you to PLAY the tape holding an RLE coded file.

LISTING #2 presents a copy of the program as I use it...part A is BASIC and does the LOAD (RLE file) and SAVE (TS2068 SCREEN$) functions, part B is the basic basic code as prepared for the TIMACHINE BASIC COMPILER (from Novelsoft), part C is the output from the TIMACHINE compiler when I compiled part B (this way you will know if you typed part B of the program in correctly). Type and save part A, type-save-and compile part B (SAVEing as directed). LOAD part A, LOAD the compiled code at location 30000 (LOAD "" CODE 30000 [ENTER], and SAVE this all to tape by typing RUN 9999 [ENTER].

Once you have "downloaded" an RLE file from CompuServe (for example), save this to tape for RLE decoding. (Use GO PICS on CompuServe to get to the RLE liabrary. I use MTERM II with [none] conversion and the DC2/DC4 download protocol from CompuServe. Save the RLE code using : SAVE "RLE.PIC" CODE 26710, xxxxx where xxxxx is the bytes-used parameter from the buffer). NOTE: CompuServe DC2/DC4 will automatically open your buffer and transmit the file when transmission stops, press ENTER. The RLE file will appear to be random letters and characters on your screen...an ASCII equivalent of the graphic code

To run the RLE decoder program, LOAD the program. It will auto-run. "Play tape to LOAD the RLE file" will be displayed. Play the downloaded RLE file. Immediately the picture will begin to form (somewhat slower in BASIC). The RLE conversion process will take a few seconds (15-20) compiled, 20-30+ minutes in BASIC! All 24 lines of the picture are shown on the screen. The last 2 lines take a few seconds to develop in the compiled program (you will think that the computer has died in the BASIC version, but be patient...once the picture is completed, a series of BEEPs will signal that the picture is done. Now you have a few options available to you at the press of a key:

PRESS C to send a COPY of the screen to the TS2040 printer.

PRESS I to invert (exchange paper and ink)*

PRESS S to SAVE the picture to tape**

*Note: RLE pictures use white ink on black paper, the 2040 printer will print ink black and paper white. So

RUN LENGTH ENCODED GRAPHICS

you must use the INVERT function to switch these before printing. Try printing both images.

**Note: When you SAVE the screen/picture, a copy of the screen is copied into high memory. This takes about one second in the compiled program, but much, much longer in BASIC. When transfer is complete, you will be asked for a file name. Then this high memory version is SAVEd; this is done to SAVE the entire picture. When you SAVE anything, the system destroys the bottom 2 lines of the screen with the SAVE messages. This process preserves these lines, and restores them to the screen when the SAVE is completed.

CompuServe has a wide variety of RLE files in their libraries. These include: New RLE uploads, Art Exhibition, Popular Faces, Sci-fi, Fantasy, Cartoons, Animal Kingdom, Japanimation, RLE Greetings, Fish and Fowl, Potpourri, and the Sysop's Favorites! The enclosed pictures were downloaded from CompuServe and are presented here with their permission: The Mermaid is an RLE copy of an art work by Maryiln Morey; the Augmented Lunar Module was provided by the Picture Support Forum. They were Decoded, then printed with the Pixel Sketch and Graphics Editor (large size mode) with an SG-10 printer.

In addition to the many RLE file libraries, CompuServe has a Picture Support Forum dedicated to online computer graphics. The forum will keep you up-to-date with bulletins, message boards, and courteous help from the system operators (sysops). In this regard, I would like to thank CompuServe and Larry Wood for the help and cooperation I got in putting this article together. Thanks Larry! Also, I would like to thank John Ryan for the use of his RLE decoder program. Now we can all enjoy with anticipation the downloading of RLE graphics, the magic of decoding the file, and watching as a work of art is painted on our computer screens! It only takes a few minutes to turn any day into Christmas again.

```
  1 REM ************************

            RLE  BASIC
    Run Length Encoded Graphics

Original program by     John Ryan
 presented with author's consent

Modified by     S D Lemke
     Lemke Software Development
          2144 White Oak
     Wichita,  Ks.     67207

*********************************
  50 DIM L$(4096): GO TO 400
 100 LET D=0
 110 IF X>255 THEN LET X=X-256:
LET Y=Y-1
 120 IF Y<0 AND Y>-17 THEN LET L
$(((ABS Y)-1)*256+X+1)="1": GO T
O 140
 125 IF Y<=-17 THEN RETURN
 130 PLOT X,Y
 140 LET D=D+1: LET X=X+1
 150 IF D=C THEN RETURN
 160 GO TO 110
 200 LET C=PEEK A-32
 210 IF C<=0 THEN RETURN
 230 RETURN
 300 LET C=PEEK A-32
 310 IF C<=0 THEN RETURN
 330 LET X=X+C: IF X>255 THEN LE
T X=X-256: LET Y=Y-1
 340 RETURN
 400 RESTORE : FOR N=24500 TO 24
505: READ X: POKE N,X: NEXT N
 410 DATA 243,6,192,195,5,10:
 REM Full Screen Copy
 420 LET P=0: LET I=7: BORDER P:
 PAPER P: INK I: CLS
 430 LET A=40000: LET X=0: LET Y
=175
 440 IF PEEK A<>71 THEN LET A=A+
1: GO TO 440
 450 IF PEEK A<>72 THEN LET A=A+
1: GO TO 450
 540 LET A=A+1
 550 GO SUB 300
 555 IF C<0 THEN GO TO 620
 560 LET A=A+1
 570 GO SUB 200
 575 IF C<0 THEN GO TO 620
 580 LET A=A+1
 590 IF C=0 THEN GO TO 550
 600 GO SUB 100
 605 IF Y<=-17 THEN GO TO 620
 610 GO TO 550
 620 PRINT #1;AT 0,0;: FOR Z=0 T
O 1
 630 FOR Y=1 TO 256 STEP 8
 640 FOR X=Z*8+1 TO Z*8+8
 650 LET D=0
 660 FOR N=0 TO 7: IF L$((X-1)*2
56+Y+N)="1" THEN GO SUB 850
 670 NEXT N
 680 POKE USR "A"+X-1-(Z*8),D
 690 NEXT X
 700 PRINT #1; PAPER P; INK I;"A
";
 710 NEXT Y
 720 NEXT Z
 730 FOR Z=1 TO 5: BEEP .1,2: BE
EP .1,1: NEXT Z: PAUSE 0: LET I$
=INKEY$
 740 IF I$="c" OR I$="C" THEN RA
NDOMIZE USR 24500: LPRINT
 750 IF INKEY$="I" OR I$="i" THE
N FOR N=16384 TO 22527: LET X=PE
EK N: POKE N,255-X: NEXT N
 760 IF I$="s" OR I$="S" THEN PR
INT #1;AT 0,0;: FOR M=0 TO 6911:
 POKE (50000+M),PEEK (16384+M):
NEXT M: GO TO 930
 800 GO TO 730
 820 FOR M=0 TO 6911: POKE (1638
4+M),PEEK (50000+M): NEXT M: GO
TO 730
 850 IF N=7 THEN LET D=D+1: RETU
RN
 860 IF N=6 THEN LET D=D+2: RETU
RN
 870 IF N=5 THEN LET D=D+4: RETU
RN
 880 IF N=4 THEN LET D=D+8: RETU
RN
 890 IF N=3 THEN LET D=D+16: RET
URN
 900 IF N=2 THEN LET D=D+32: RET
URN
 910 IF N=1 THEN LET D=D+64: RET
URN
 920 IF N=0 THEN LET D=D+128: RE
TURN
 930 GO TO 1200
 940 STOP
1010 REM        DECODE.RLE
1020 CLEAR 39999
1090 REM        SAVE
1100 CLS : PRINT "Play your TAPE
 to LOAD RLE file.": LOAD ""CODE
 40000
1110 GO TO 50
1190 REM        SAVE
1200 INPUT #0;AT 0,0;"File Name
= "; LINE n$: SAVE n$CODE 50000,
6912
1210 GO TO 820
9999 SAVE "RLE.DECODE" LINE 1020
```

Listing #1

NOTE: Lines 680 and 700, the "A" is UDG A, GRAPHICS A.

Listing #2-A

```
1005 REM ************************

         RLE -- DECODER

RLE SAVE/LOAD program...
used with the RLE Compiled BASIC

********************************

1010 REM        DECODE.RLE
1020 CLEAR 39999: LOAD "RLE.DECO
DE"CODE 30000
1090 REM        LOAD
1100 CLS : PRINT "Play your TAPE
 to LOAD RLE file.": LOAD ""CODE
 40000
1110 RANDOMIZE USR 30000
1190 REM        SAVE
1200 INPUT #0;AT 0,0;"File Name
= "; LINE n$: SAVE n$CODE 50000,
6912
1210 RANDOMIZE USR 31265
1220 GO TO 1200
9999 SAVE "RLE.DECODE" LINE 1020
: SAVE "RLE.DECODE"CODE 30000,19
70
```

Listing #2-B

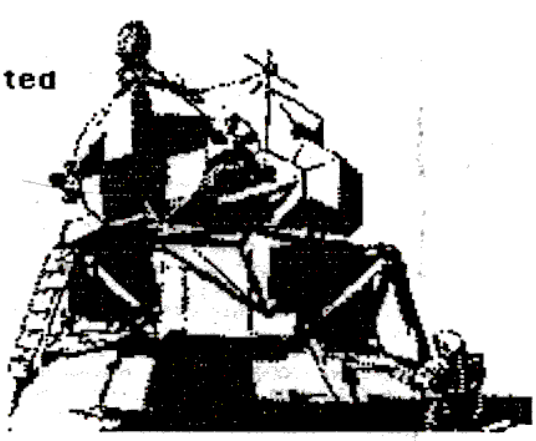
```
   1 REM ************************

       Compiled RLE  BASIC
  Run Length Encoded Graphics

Original program by     John Ryan
 presented with author's consent

Modified by     S D Lemke
     Lemke Software Development
          2144 White Oak
     Wichita,  Ks.     67207

********************************
   5 REM !USR 30000
  10 REM !INT C,X,Y
  20 REM !INT +A,D,I,N,P,Z,M
  25 REM !LEN I$<=1
  30 REM ! LPRINT
  35 REM ! LIST
  40 REM ! OPEN #
  50 DIM L$(4096): GO TO 400
 100 LET D=0
 110 IF X>255 THEN LET X=X-256:
LET Y=Y-1
 120 IF Y<0 AND Y>-17 THEN LET L
$(((ABS Y)-1)*256+X+1)="1": GO T
O 140
 125 IF Y<=-17 THEN RETURN
 130 PLOT X,Y
 140 LET D=D+1: LET X=X+1
 150 IF D=C THEN RETURN
 160 GO TO 110
 200 LET C=PEEK A-32
 210 IF C<=0 THEN RETURN
 230 RETURN
 300 LET C=PEEK A-32
 310 IF C<=0 THEN RETURN
 330 LET X=X+C: IF X>255 THEN LE
T X=X-256: LET Y=Y-1
 340 RETURN
 400 RESTORE : FOR N=24500 TO 24
505: READ X: POKE N,X: NEXT N
 410 DATA INT 243,6,192,195,5,10
:       REM Full Screen Copy
 420 LET P=0: LET I=7: BORDER P:
 PAPER P: INK I: CLS
 430 LET A=40000: LET X=0: LET Y
=175
 440 IF PEEK A<>71 THEN LET A=A+
1: GO TO 440
 450 IF PEEK A<>72 THEN LET A=A+
1: GO TO 450
 540 LET A=A+1
 550 GO SUB 300
 555 IF C<0 THEN GO TO 620
 560 LET A=A+1
 570 GO SUB 200
 575 IF C<0 THEN GO TO 620
 580 LET A=A+1
 590 IF C=0 THEN GO TO 550
 600 GO SUB 100
 605 IF Y<=-17 THEN GO TO 620
 610 GO TO 550
 620 PRINT #1;AT 0,0;: FOR Z=0 T
O 1
 630 FOR Y=1 TO 256 STEP 8
 640 FOR X=Z*8+1 TO Z*8+8
 650 LET D=0
 660 FOR N=0 TO 7: IF L$((X-1)*2
56+Y+N)="1" THEN GO SUB 850
 670 NEXT N
 680 POKE USR "A"+X-1-(Z*8),D
 690 NEXT X
 700 PRINT #1; PAPER P; INK I;"A
";
 710 NEXT Y
 720 NEXT Z
 730 FOR Z=1 TO 5: BEEP .1,2: BE
EP .1,1: NEXT Z: PAUSE 0: LET I$
=INKEY$
 740 IF I$="c" OR I$="C" THEN RA
NDOMIZE USR 24500: LPRINT
 750 IF INKEY$="I" OR I$="i" THE
N FOR N=16384 TO 22527: LET X=PE
EK N: POKE N,255-X: NEXT N
 760 IF I$="s" OR I$="S" THEN PR
INT #1;AT 0,0;: FOR M=0 TO 6911:
 POKE (50000+M),PEEK (16384+M):
NEXT M: GO TO 930
 800 GO TO 730
 810 REM ! OPEN #
 820 FOR M=0 TO 6911: POKE (1638
4+M),PEEK (50000+M): NEXT M: GO
TO 730
 850 IF N=7 THEN LET D=D+1: RETU
RN
 860 IF N=6 THEN LET D=D+2: RETU
RN
 870 IF N=5 THEN LET D=D+4: RETU
RN
 880 IF N=4 THEN LET D=D+8: RETU
RN
 890 IF N=3 THEN LET D=D+16: RET
URN
 900 IF N=2 THEN LET D=D+32: RET
URN
 910 IF N=1 THEN LET D=D+64: RET
URN
 920 IF N=0 THEN LET D=D+128: RE
TURN
 930 REM ! CLOSE #
 940 STOP

NOTE: Lines 680 and 700, the "A"
 is UDG A, GRAPHICS A.
```

Listing #2-C

```
  LINE 40:  +0
  LINE 810:  +1265
  LINE 40:  30000 #7530
  LINE 810:  31265 #7A21

 RT1       31535 #7B2F
 RT2       31536 #7B30
 RT4       31545 #7B39
 RT8       31554 #7B42
 RT9       31555 #7B43
 RT10      31564 #7B4C
 RT11      31573 #7B55
 RT12      31574 #7B56
 RT13      31583 #7B5F
 RT14      31589 #7B65
 RT15      31590 #7B66
 RT16      31599 #7B6F
 RT20      31605 #7B75
```

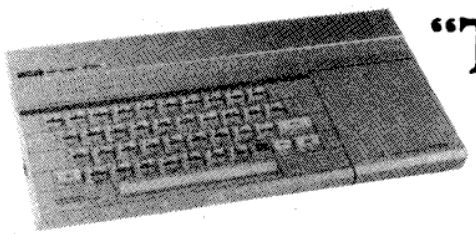
Continued Next Page.

Augmented Lunar Module



```
RT21      31613 #7B7D
RT22      31621 #7B85
RT28      31629 #7B8D
RT29      31631 #7B8F
RT33      31691 #7BCB
RT48      31704 #7BD8
RT50      31716 #7BE4
RT55      31728 #7BF0
RT61      31737 #7BF9
RT62      31781 #7C25
RT63      31792 #7C30
RT69      31809 #7C41
RT70      31825 #7C51
RT71      31833 #7C59
RT89      31839 #7C5F
RT93      31850 #7C6A
RT105     31856 #7C70
RT110     31866 #7C7A
RT111     31871 #7C7F
RT114     31879 #7C87
RT116     31890 #7C92
RT117     31896 #7C98
RT122     31902 #7C9E
RT124     31910 #7CA6
RT125     31915 #7CAB
RT126     31920 #7CB0
```

```
RT128     31927 #7CB7
c.......... INTEG     31970 #7CE2
x.......... INTEG     31972 #7CE4
y.......... INTEG     31974 #7CE6
a.......... POSINT    31976 #7CE8
d.......... POSINT    31978 #7CEA
i.......... POSINT    31980 #7CEC
n.......... POSINT    31982 #7CEE
p.......... POSINT    31984 #7CF0
z.......... POSINT    31986 #7CF2
m.......... POSINT    31988 #7CF4
i$         STR(1)     32010 #7D0A
l$(4096)   STR        32013 #7D0D

TIME MACHINE ©1986 Cameron Hayne

M/C: 1970 BYTES
+ 4141 BYTES FOR M/C VARIABLES
(BASIC WAS 2380 BYTES)

SAVE "m/c"CODE 59257,1970
LOAD "m/c"CODE 30000
```



"The Mystery of the Missing 253"

Part Four

By Wes Brzozowski

We stopped last time in mid-description. It seems our Good Editor has such a wealth of excellent contributions last issue (besides mine, I mean) that my article had to be cut to fit the space remaining. This is just fine; my volumninous verbage often takes up so much space that I worry that I am pushing many worthy authors from these pages.

I don't have to worry about that anymore.

In any case, we'll just continue where we left off last time. We were discussing flowchart 4, which shows the building of the SYSCON table. You'll want to turn back to the last issue, read the final 4 paragraphs of the last installment, and continue here (back issues are available)...

We know that expansion banks could control extra hardware (printers, disk drives, ect.), but they also could have caused an interrupt, by grounding the INT line on the backplane, with an Open Collector driver.

The subject of interrupts is far outside the scope of this series, but they effectively cause a special subroutine to be run due to an external hardware signal, rather than the execution of a CALL statement. In the standard TS2068, an interrupt occurs every 1/60 second, causing the keyboard scanner to be run. Depending on the hardware causing the interrupt, it may need a fast response, or it may be prepared to wait all day to be serviced.

A bank switching interrupt handler would have had to poll each bank to determine who caused the interrupt (or wether it was just a request to scan the keyboard) and this takes time. The ability to renumber banks according to an interrupt priority would ensure that the banks that must be serviced fastest also have the lowest bank numbers. This makes it easy to check the critical ones, first.

Now suppose that when we installed the bank number, the bank really DID exist. We put the bank number into SYSCON 01 for that bank, but with bit 7 also set. This flag states that this bank hasn't been renumbered yet.

Starting with X0A58, we try several things at once. The bank can be either RAM or ROM bank. The best way to check for RAM is to write a number, and then read it back, to see if it "took". Of course a ROM bank just MIGHT have the same value by coincidence, so it pays to write a second value to the same location, and check it again.

Unfortunately, as we can see from the flowchart, the routine to move bytes from one bank to another has been misapplied; it errantly tries to copy bytes from the RAM bank to the EXROM bank. It's obvious that the folks at Timex didn't have any expansion bank hardware, at least when the code for the ROM was frozen. These are the simple, "preliminary" mistakes a programmer makes when writing code in a hurry, and prior to debugging. If any hardware were available to check this code out, these problems would never have existed.

Another thing that's done (properly this time) is to move a 24 byte block from location 0000 of the expansion bank into SYSCON 2 and the bytes following. If it's a RAM bank, this is harmless "power up" garbage. If it's a ROM bank, then these are the overhead bytes for the bank, and they'll follow a pattern like the SYSCON Table Configuration, given in part 2. (Note that a couple of items got left out of that table; we'll put in corrections as needed.)

Having done all this, we execute code to do the final setup for the bank. If it's a RAM bank, we end up at X0AB7, where we CALL a routine to check which chunks of the bank actually contain RAM. (It was also supposed to copy in the interrupt handler from the EXROM, but if fouls that up.) In any case, SYSCON 02 gets a byte whose bits specify the chunks where RAM is available.

If it's instead a ROM bank, we reset bit 5 of SYSCON 02. My SYSCON configuration could use a tad of clarification on this point. This could have been an ASCII character representing a channel specification. This is because the bank switching system could have allowed for additional channels, which could have been linked to an expansion bank. We'll talk more about I/O in a minute, but for now, we'll just point out that resetting the bit will shift the character to upper case.

Now, at X0AC7 and following, we CALL a routine to check SYSCON 15 for the bank, and if it contains 01, we get the address of the initialization code from SYSCON 07 and 08, and run the code at that address in the expansion bank. Thus, for a little while, the expansion bank is in control of the system. There are lots of things we may want to do with this. There are cases where it would be more convenient to permanently add a channel, without using the OPEN# routine (this should become clearer, in a bit) or, you may wish for it to install some special code in the Home Bank. It could even take over the entire system, or prompt you as to whether you'd like it to do so.

Whether it was RAM or ROM, initialize or not, we always end up at X0ACA, which steps the daisychain to the next bank, and we loop to X0A4C and try to find another bank.

Now that we've seen how the TS2068 builds the SYSCON table, we can take a better look at Flowchart 3, which actually installs the bank number. First let's note that the system variable MAXBNK normally contains the number of expansion banks plugged into your system. But during initialization, it's the number of the bank presently being assigned. Since the very last bank number assigned equals the number of expansion banks, everything works out nicely.

We start by assuming that there's another bank to initialize and increment MAXBNK. If it turns out we're wrong, we'll correct it later. At X0BDB we try to install that number in the next bank, which is selected by the Daisychain. That bank now has the number (MAXBNK). By sending that number to register 80 (Bank Number Access) we can access its Horizontal Select Register.

At X0BE7, we send 00 to register 40 (Horizontal Select). This does not remove "power on garbage" as the flowchart says; at power on, the bank resets itself. However, this DOES make this routine useful to another (presently unused) routine in the EXROM that RESETS the SYSCON table, by fixing up any values that

we may have changed there. It's at X0C4C; check it out and see if you can find some use for it. It appears that Timex may have once had plans to access this through one form of the RESET command, from BASIC. We'll talk about that next time.

In any case, at X0BEE, we save the maximum bank number, the contents of location A000, and then write 04 there; this is NOT an unlock command, as the flowchart says. One of the bank's status registers (the A0 register) is memory mapped into location A000. But we don't yet know if the bank we're setting up actually exists, yet! If it does, then when we read register A0, we'll find bit 2 = 0. BUT IF IT'S NOT THERE, WE'LL JUST GET THE CONTENTS OF LOCATION A000 WHEN WE TRY TO READ IT. As such, we first set the contents of A000 so that bit 2 = 1. If the bank doesn't exist, we are guaranteed to see a "1" there.

Almost. What if NO bank has location A000's chunk allocated to it? This can happen, since that routine that resets the SYSCON table also CALLs this, and the chunk could have been "lost" through some code of our own. Looking at the TS2068 schematic, we see that data line D2, and only D2, has a pullup resistor on it. Even if no bank will respond to location A000, we're still covered, and the lack of an expansion bank will show us a 1. This would also have been needed if Timex sold versions of the TS2068 that only had 16k of RAM in them. (They did announce such plans, though they wisely discarded them.) This is a somewhat more complete (and slightly more accurate) explanation of the resistor than was given in the past. If this explanation makes any sense, you may see why I simplified the description, earlier.

All right, NOW we can read the C0/A0 register pair, and check bit 2 to see if the bank exists. In either case, we'll restore the contents of location A000, that we wiped out earlier. If the bank is there, we set the CY flag, and return to the CALLing routine with the bank's number installed, and MAXBNK properly updated. If the bank isn't there, we return with the CY flag reset, we decrement MXBNK, to correct our original assumption that another bank existed, and we send 04 to register C0, to end the setup mode, since we won't be using the daisychain, any more. That's it!

Flowchart 5 shows the GET_STATUS routine, in the RAM resident code, after the modifications in TM6.5.2 have been installed. For a specified bank, it will return the Horizontal select byte, and will also return the status byte for an expansion bank. Note that in normal use, this routine is called once for each bank, and the information is used as a whole. This is because the Horizontal Select register for the standard banks "claims" all 8 chunks for those banks. Remember, an expansion bank has to override this, by applying the $\overline{BE}$ signal at the computer's backplane connector. As such, the horizontal select information for the standard banks is only valid for those chunks not claimed by an expansion bank.

There's not too much to say about Flowchart 6. This is CALLed when the initialization code finds a ROM bank. It marks it as such in the SYSCON table, and checks bit 0 of SYSCON 15. If it's a "1", then the initialization code for the bank is run. This allows each bank the option of participating in system initialization. It's not mandatory, but it's nice if it's needed; particularly if the bank has some I/O hardware that needs some initial massaging. Note that HL is used throughout the initialization as an address pointer into the SYSCON table. As the flowchart shows, this routine has a major bug in that it wipes out that pointer by accident. This does not seem difficult to fix, but as is, it seems unlikely that the system could initialize with a ROM bank present.

Increasing Your Vocabulary

Many readers know about the working TS2068 commands that aren't documented in your owner's manual. For example, OPEN #2, "p" will redirect all output from a PRINT statement to your printer, rather than your screen. There are also commands that are only "half there". Turn on your TS2068, and type in the following "program":

```
10 LOAD *"m",3,"test"

20 CAT "d" ,3,4

30 FORMAT "m",1,2,3,"junk"

40 OPEN #3,"j",1,2,"moretrash"

50 MOVE "a" ,"garbage",2,5

60 ERASE "b" ,1,"nonsense"
```

You may be surprised to find that every one of these commands can be entered into your machine, and it will accept them, but not one of them will actually RUN! (You'll get an error message, instead.) Furthermore, each one will take as long a list of string and numberic items as you'd like to give, provided you give at least one, following the single letter in quotes. (Except for the OPEN# command, which normally needs no extra list following the letter.

What gives? The Timex/Sinclair machines are supposed to do complete syntax checking when you type your lines in; how did it miss these? Well, there is a class of commands that only work when extra hardware is plugged into your machine. There are two ways these could have been implemented, and the TS2068 designers seem to have left both options open. The first method is largely copied from the Sinclair Spectrum, and it works like this. BASIC can do two things when it "sees" a program line. If you're typing the line in, it runs the Syntax Checker. If it's RUNning a program, it looks up the address of the routine that executes the command and runs it. (If you type in a line with no line number, it does both.)

This is also true for the above extended commands. We think we see a difference because the routine that runs the command is designed to end up with the printing of an error message. Thus, if you type in the proper syntax, the machine will properly accept the line, and when you RUN it, it "properly" prints an error code.

Whatever for? The program that prints error messages (for both the TS2068 and the Spectrum) is at location 0008. Those familiar with Sinclair's Interface One, for the Spectrum, know that it switches in it's "shadow ROM" whenever the instruction at 0008 is run. The shadow ROM then checks the cause of the error, scans the present BASIC line, determines if it's supposed to be running an extended command, and acts accordingly. While this might seem like an odd way to add commands, it contains a perverse sort of beauty. It makes it possible to design a computer and include all the ROM code necessary to run future add-ons, without really knowing what those add-ons will look like, or what real software is needed to run them. It's a great way to "buy time".

If this method were used, we can guess that the BEU would have contained extra hardware to switch in the "Superbank" mentioned in the past. This would be analagous to the shadow ROM. Since the Home ROM code contains nothing to link it to a bank switching interrupt handler, perhaps the "Superbank" may have switched in when the code at 0038 (the keyboard interrupt handler) was run.

There is a second option. In routines to "run" the extended commands, we tend to find a JUMP instruction to code to print the error message. But following that JUMP is usually found extra code that appears to look up an address in the SYSCON table and CALL the routine in its expansion bank. It also passes on whatever list of information was tacked onto the end of the statement. As such, if the JUMP is NOPed out, it appears that the system should find code in an expansion bank to actually handle the command. Furthermore, these extra blocks of code are not used anywhere else in the ROM! They were almost certainly intended to link the extended commands to the expansion banks. Why were they blocked from that purpose?

Nothing is simple. Once again, there are two fairly reasonable options. In essentially every case, the little packet of code that's blocked off contains one or more fatal bugs that could really gum up the system if allowed to run. Since it would have been fairly clear to the designers that new, bugless ROMs would be needed for bank switching anyway, they could save debug time by simply hiding the code that the original ROMs would really not need.

The other option centers around the timing of separate Timex and Sinclair developments. The Sinclair Interface One was released around the same time as the TS2068 was, and its relative simplicity suggests that design on the TS2068 was begun a good deal BEFORE the Interface One. As such, the original TS2068 designs could not have considered it, and if Timex did eventually plan to copy the Interface One's method into its own microdrive interface, they would have had to make some changes.

While blocking off some of their code might seem a sloppy way to do this, it would have worked, and the rest of the Timex modifications to the Spectrum code aren't very neat, either. As a glaring example, we can find several routines in the ROM that were probably used by the programmers to debug the code, but aren't used by the ROM, itself. This method is fairly universal, but the common practice is to remove your debug garbage before assembling the version that's to go into ROM.

It's a similar bit of sloppiness that makes this second option the most likely. You see, one of these "blocked off" bits of code seems to have quite a few instructions missing from it; it could never work as is. Now, I know that some of you have bootleg copies of Timex's original source code listing for the

Continued Next Page.

ROMs. If you'll look at the code following the JP at 25E1 in the Home ROM, you'll see that Timex "commented out" a full 28 lines of code, which would have assembled into about 43 bytes. These would have restored the missing functions, but the Home ROM only has 36 spare bytes in it (3CDC, and following), so these extra bytes wouldn't have fit. Rather than to search for debug garbage to delete, they simply hacked out some code that might otherwise have been functional! Clearly, it wasn't too important to them. As such, they were probably going to copy the method (and as much software as possible!) from the Interface One.

While we can bounce these, and a whole lot of other bits of circumstantial evidence around, we can get no conclusive answer. In the end, it doesn't matter. If we wish to restore the bank switching functions, we can use either option. But it does help to understand that both options are there. It's also worthwhile to note that restoring the blocked out code would make the hardware design somewhat simpler for us.

I/O, I/O, It's off to Work We Go...

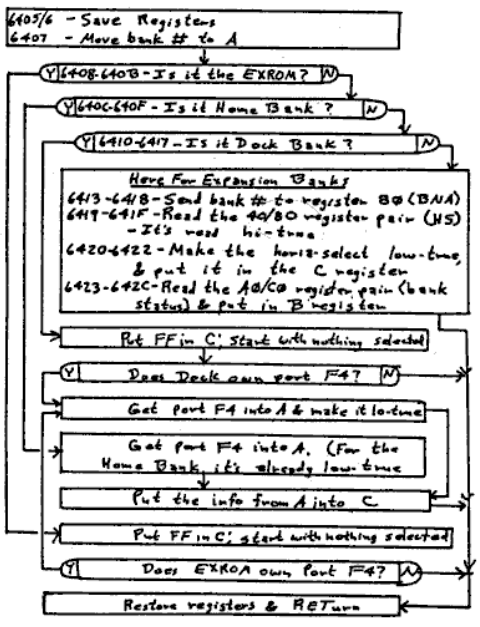
Take some time and get cozy with TM4.1, on I/O channels. This is not a great treatment of the subject, but it's a start. Next, if you'll read the definitions of the system variables STRMS and CHANS, on pages 262, 263 of your TS2068 User Manual, you'll notice some subtle inconsistencies with the Technical Manual. The User Manual implies that channels and streams are two different things, and that channels are "attached" to streams. The Technical Manual suggests that the two things are identical.

Actually, a channel is a block of information providing a link to an I/O device. At a minimum, it contains a 2 byte output address for the device, a 2 byte input address, and a 1 byte device specification, which is an ASCII character. All of the normal channels that appear when you power up your computer, ("k", for keyboard & lower screen, "s", for main screen, "p", for printer, and "r", which isn't used, but is there anyway) follow this 5 byte format. It doesn't have to be this way though; an "m" (microdrive) channel on the Spectrum, is an incredible 595 bytes long!

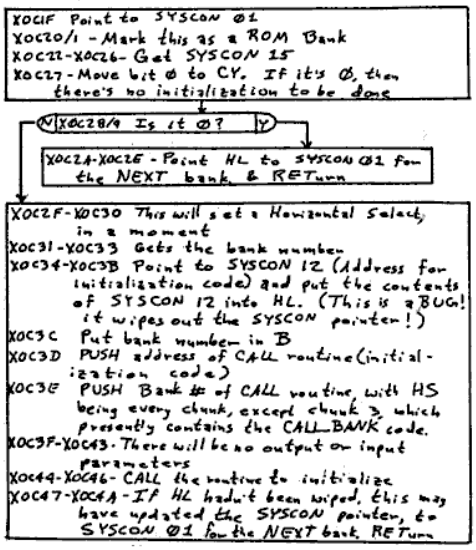
A stream is normally a displacement into the channel area. There are 19 available streams, and the system does most of it's I/O through them. It must look up the channels they point to, find the addresses of the appropriate input or output routines, and then jump to them. Normally, stream 0 points to the "k" channel, stream 2 to the "s" channel, and stream 3 to the "p" channel.

Can we use this from BASIC? Sure thing! If you type:

```
PRINT #0;"test":PAUSE 0
```



FLOWCHART 5. GET_STATUS, in RAM resident code. Incorporates the corrections in TM6.5.2 CAUTION: This routine is intended to be CALLed once for each bank. The Home bank will appear to "own" chunks allocated to expansion banks, and this will be misleading without the extra information on the expansion banks.



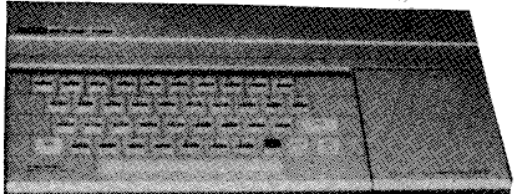
FLOWCHART 6. Checks to see if a ROM bank needs to be initialized. If so, it executes the bank's initialization routine through the use of CALL_BANK.

you'll find that it prints on the bottom line of the screen, where BASIC can't usually PRINT. The PAUSE 0 is simply there to keep the system from printing its "OK" down there before you can see what you printed. If you instead use #2, it will PRINT normally on the screen. Using #3 will send the information to the printer. What we're doing is telling BASIC which stream to use when it sends out the PRINT data. When we don't give it a stream number, it uses #2, as a default value.

Conversely, we can do the same thing with the INPUT command. The command INPUT #1,A will input a character through stream 1. This is what it does by default, and so doesn't demonstrate as much as we'd like, but it shows how we'd use INPUT to take data directly into BASIC from an I/O device. None of the other channels has a true "input address": the addresses given will just cause the printing of an error message. Clearly, there's a lot of I/O power here that just isn't being used!

The "print drivers' that allow LPRINT and LLIST commands to run a large printer work because they modify the output address in the "p" channel. (Ordinarily, it points to a routine in the ROM that accommodates the TS2040 printer.) The COPY command needs a separate routine because the COPY command doesn't work through an I/O stream.

Most of this can be gleaned from the Technical Manual, but there are additional capabilities that have not been documented. Above, I said that a stream is NORMALLY a displacement into the channels area. Actually, only the lower 15 bits of the value are a displacement. (The most significant bit is then normally "0".) However, if the most significant bit contains a "1", then the other bits represent a displacement into the SYSCON table, and can give us an I/O link to routines in an expansion bank.



Some machine code programmers use the RST 10 command to print the character in the A register on the screen. Actually, RST 10 will send it to the "current channel" (whose address is in the system variable CURCHL), and BASIC will have set this to the "s" channel somewhat before it executes the USR funtion that hands control to our machine code. It can be changed by putting a stream number in A and CALLing 1230. For example, stream 3 normally points to the "p" (printer) channel. If we LD A,3 and then CAL 1230 then subsequent RST 10 commands will send the character in A to the printer.

Now, this RST 10 business is standard Spectrum stuff, but Timex added a lot more for bank switching (mostly inoperable, due to bugs). There is also a "current channel bank number", in the system variable CURCBN. For expansion banks, this is the bank number, and for Home Bank, it's set to 0. The Dock and EXROM banks aren't supported by this. As such, RST 10 was intended to be able to send the value in the A register to a routine in any expansion bank. If it were'nt for a bug, it could also be used to INPUT a character through a routine in an expansion bank numbered 2 or greater. Apparently, Timex had a special purpose in mind for bank #1 (the superbank, perhaps?).

If you're comfortable with streams and channels, you probably realize that the primary function of the OPEN # command in the standard TS2068 is to modify a stream to point to a particular channel (OPEN #stream,"channel"). However, there is also a "channel specific" portion run, since there may also be some system flags that need massaging. To make it possible to OPEN a channel into an expansion bank, placing a comma after the standard OPEN # format will allow you to add any additional garbage you'd like to the line; it needn't be a list at all. This will pass the syntax check, but trigger an error on execution, kicking in the superbank, (if that method were used) and handling whatever channel specific operations may be needed. Like actually inserting the new channel. Or scanning the SYSCON table for the proper channel specifier and running a routine to open the channel from that bank. The OPEN # code address would be at SYSCON 03 & 04; this got left out of the table, in part 2.)

The CLOSE # command looks a bit more boring, but it does a lot. In the standard TS2068, it largely just returns to its power on value, but if the stream was attached to an expansion bank channel, it will also run some code form that bank. The address is found at SYSCON 05 and 06 this WAS included in the table in part 2. (Well, every now and then, SOMETHING goes right!) In order to get some use out of extended bank switching, the I/O routines must be understood and debugged. This is a bit far from the topic of this series, and space won't allow a detailed examination, but here are some memory addresses to help (keep your bug- spray handy!!!)

11AA-11BE Initial Channel Data

11C1-11CD Initial Stream Data

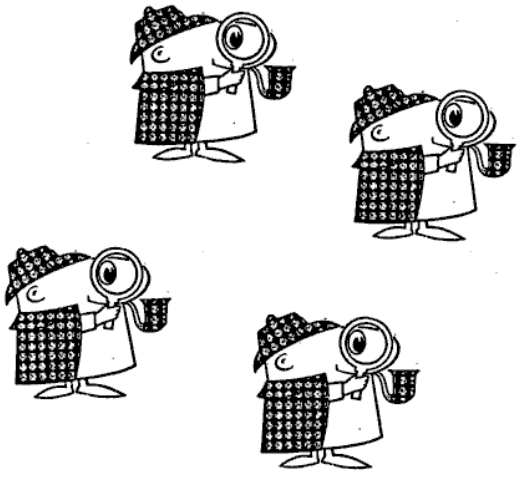
11ED-122F Outputs A to current channel (used by RST 10)

1230-1292 Set current channel according to stream # in A

1374-139E Search SYSCON table for channel specifier in C

139F-1429 CLOSE routines

142A-14C6 OPEN routines - Note that location 1486 contains a JR that is reached through another JR. This second JR is one of those JUMPS that blocks off some of the ROM code. depending on how you may want to implement things, this JR might be NOPed to allow OPENing a stream through an expansion bank.



"I really meant it... I really did"

I began this series with the cautionary note that I'd be presenting only the results of my foray through the ROMs; not giving a construction project. But perhaps I can break my own rule just this once. Some readers are a bit scared by the idea of changing the ROM code, to make the bank switching work properly. Cutting up their computers, and opening it repeatedly to switch EPROMs just seems too bothersome. Actually, there's a better way, which is so simple that it is by far the easiest part of implementing bank switching.

Figure 5 shows a circuit I use in order to run EPROMs in place of the ROMs. I was able to build mine on a small card that plugs into the cartridge slot, although it's slightly too large to get the door closed. This is not really too bothersome, since it's only a temporary modification, used when debugging the actual ROM code. One fly in the ointment is that one necessary signal is not available on either connector, and you'll have to open up your computer to tack a wire on to it.

The jumper marked W2 is the point to which the wire must be attached. Although it looks like a resistor, it's merely a wire jumper in disguise, so you can hook the wire to either side. While you've got your computer open, don't forget to remove the ROMs. It's also a good idea to put a label on each, telling which is which; if you ever want to put them back, you'll need to know.

Depending on the installation of the jumpers as shown in Figure 5, you can run either EPROMs or the original ROMs, or one of each. This is helpful in debugging the board, and also in debugging your ROM modifications, since you may frequently want to switch back to the original ROMs, to see how they react to a certain set of circumstances. I find it most convenient to keep a set of EPROMs that contain an exact copy of the ROM code for this. That way, I don't have to fool with the jumpers, much. Still, you can do whatever suits you best. Also note that the extra wire is only needed if you want to make changes to the Home ROM code. You can simulate EXROM externally without any extra wires tacked on, but you'll still want to open the computer and remove the real EXROM, first.

Since you'll be doing a lot of plugging and unplugging, invest the extra ten or fifteen bucks to install ZIF sockets. This will be cheaper in the long run, since you can't plug a chip too many times into a normal socket, without breaking off a pin. The cost of the sockets will far outweigh the cost of the ruined EPROMs, not to mention the wasted time and frustration.

That's all for now; we'll wrap up this series next time. Don't forget to write or call with your questions, ideas or observations. I'll be glad to hear from you!

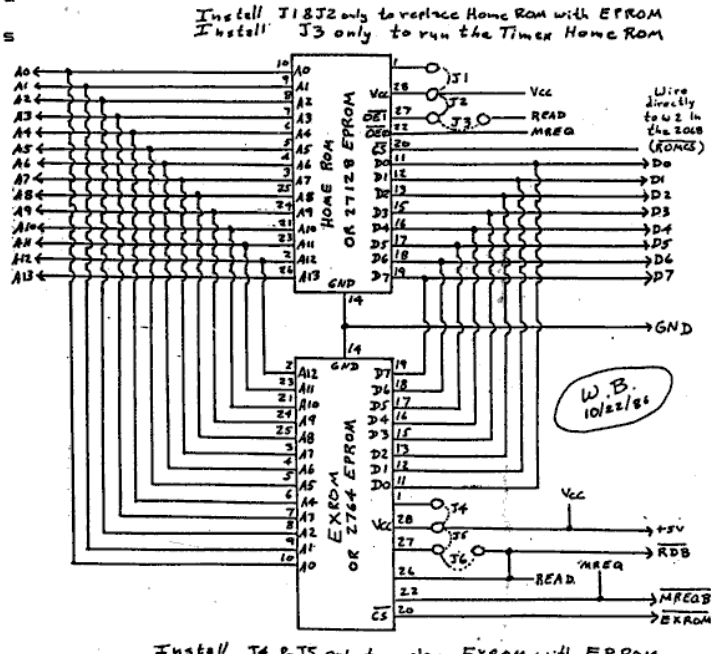


FIGURE 5. An external circuit board for debugging changes in Home ROM and EXROM code

CHARACTER ANALYSIS

By William C. Andrews

CHARACTER ANALYSIS is a utility program to define the binary and decimal value of a byte and is useful with user defined graphics when they appear in a machine code program. It can be merged with another program and activated with GOSUB 9600.

```
9600 BORDER 5: PAPER 5: CLS
9610 CLS : PRINT AT 10,0;" ENTER
CHARACTER TO BE ANALYZED "
9620 INPUT "(GRAPHIC MODE) ";A$
9630 CLS
9640 GO SUB 9880
9650 PRINT AT 4,15; PAPER 7; BRIG
HT 0;A$
9660 LET x=120
9670 LET l=9
9680 LET y=143
9690 FOR a=y TO y-7 STEP -1
9700 LET c=9
9710 LET z=7
9720 LET byte=0
9730 FOR b=x TO x+7
9740 LET byte=byte+2^z*(POINT (b,
a))
9750 PRINT AT l,c;POINT (b,a)
9760 IF POINT (b,a)=0 THEN  PRINT
 AT l,c; OVER 1; PAPER 6; BRIGHT
1;" "
9770 IF POINT (b,a)=1 THEN  PRINT
 AT l,c; OVER 1; INVERSE 1; BRIGH
T 1;" "
9780 PRINT  PAPER 1; INK 7; BRIGH
T 1;AT l,19;"      ";AT l,20;byte
9790 LET c=c+1
9800 LET z=z-1
9810 NEXT b
9820 LET l=l+1
9830 NEXT a
9840 PRINT  PAPER 2; INK 9; BRIGH
T 1;AT 18,9;" BINARY ";AT 18,17;
PAPER 1;" DECIMAL "
9850 PRINT #1;TAB 10;"COPY? (Y/N)
"
9860 PAUSE 0: IF INKEY$="y" THEN
 COPY
9870 CLS : PRINT AT 10,0;"REMEMBE
R TO CHANGE GRAPHIC MODE": PAUSE
200: CLS : STOP
9880 PLOT 111,152
9890 DRAW 25,0: DRAW 0,-25: DRAW
-25,0: DRAW 0,25
9900 PRINT AT 3,14; PAPER 7;"    "
9910 PRINT AT 4,14; PAPER 7;"    "
9920 PRINT AT 5,14; PAPER 7;"    "
9930 RETURN
```

DUNGEON OF YMIR

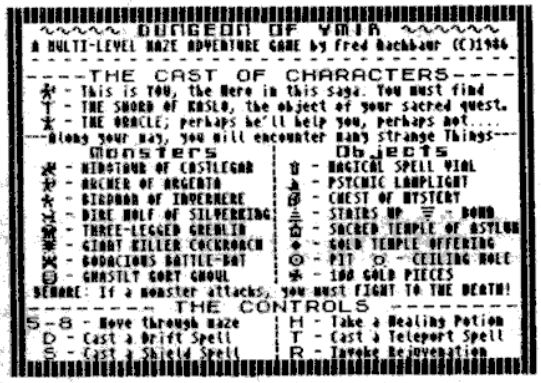
A MULTI-LEVEL MAZE ADVENTURE GAME by Fred Nachbaur (C)1986

FOR THE TIMEX TS1500

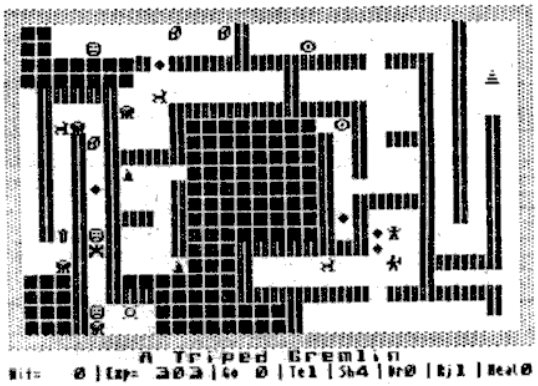
FINALLY! A FULL-FEATURE, HIGH RESOLUTION DUNGEON GAME FOR THE TS1500!

This 24K game, written entirely in machine-code, is the most spectacular program ever written for the TS1500. Nine levels, 16 types of monsters, 14 objects, six spells. Easy to play, difficult to master. Includes FAST-SAVE with auto-boot to save in-progress games: time to load entire program reduced to 70 seconds! Revolutionary TRUE HI-RES puts your TS1500 on a par with much larger machines.

Send $24.95 (cheque or MO) to FRED NACHBAUR, C-12 MTN. STN. GROUP BOX, NELSON BC V1L 5P1 CANADA. Specify version: V1 (TS1500 + 8K Hunter NVM or equivalent) or V2 (TS1500 + 16K RAM pack). V2 requires a minor hardware addition (included). COMING SOON: V3 for ZX81/TS1000. Inquire. *** ALSO AVAILABLE: TS1500 HI*RES EXTENDED BASIC ($16.95)



ALL ARTWORK IN THIS AD FROM ACTUAL SCREEN DUMPS!



Sharp's Inc.
Rt. 10, Box 459
Mechanicsville, VA 23111
(804) 746-1664 or 730-9697

COMPARE OUR PRICES!

QSOUND/QPRINT A sound generator board	$129.95	METACOMCO:	
THE EDITOR 10 times better than Quill	$34.95	PASCAL	$79.95
		BCPL	$69.95
512 K RAM	$164.95	APL	$99.95
DISK I/F	$124.95	LATTICE 'C'	$99.95
QFLASH	$29.95	LISP	$69.95
512K & DISC I/F	$289.95	ASSEMBLER	$45.95
CENTRONICS I/F	$34.95	BOOKS:	$8.95 EACH
SUPERCHARGE	$69.95	MACHINE CODE PROG.	
AQUANAUT 471	$25.95	QDDS COMPANION	
THE KING	$25.95	QL GAMESMASTER	
QL PAINT	$34.95	INSIDE THE SINCLAIR QL	
SPRITE GENERATOR	$24.95	USING GRAPHICS ON THE QL	
QL MOUSE	$129.95	ADVANCED PROGRAMMING ON QL	
SUPER ASTROLOGER	$29.95	DATABASE MANAGMENT QL	
WAR IN THE EAST	$27.95	QL SUPERBASIC	
SUPER TOOLKIT II	$54.95	PROFITING FROM THE QL	
QL PRINTER	$219.00	QL COMPUTING	
ROM DISASSEMBLER	$34.95	(MANY MORE TITLES)	

MICRODRIVES IN STOCK AT $9.95 FOR (4)

VISA & MASTERCARD ACCEPTED WITH 3% SURCHARGE. ALL PRICES INCLUDE SHIPPING.
WRITE FOR OUR NEW 6th EDITION CATALOG.

TASWORD TWO +

By Bill Ferrebee

```
Tasword Two+
© Tasman Software 1983
© Mountaineer Software  1986

PRINT OPTIONS
Just press ENTER for default
values given in brackets

Line spacing? (1)
Start at line? (1)
Finish at line? (last)
No. of copies (1)
Left margin (0)
```

I use my TS2068 for many tasks. But if I had to choose one major use, I would have to say it's WORD PROCESSING. Be it for writing letters, making a check list, or compiling articles (like this one), I make great use of my Word Processor.

A short time back, I did an extensive overview of available 2068 Word Processing software for a newsletter. Of all of the programs I looked at, two were exceptional: MSCRIPT and TASWORD TWO.

Both programs were well written, and had some very nice features not found in programs for other computers.

I personally prefer using TASWORD TWO, mainly because of its screen presentation. If, for example, a line of text needs to be centered on the page, it will be centered on the screen. "What you see is what you get" is the best way to describe it.

As I said before, both programs were well written. But do we T/S users EVER leave a good program alone? Heavens, NO! We have to make it BETTER!

In the case of MSCRIPT, Jack Dohany has authored a major rewrite: MSCRIPT Version 5. Jack has added every imaginable feature, and a few you may not have thought of. All owners of the original MSCRIPT should contact Jack about his upgrade. You need to legally own a copy of the original, because he only includes documentation on the added new features. Write to: Jack Dohany, 390 Rutherford Ave., Redwood City, CA 94061.

As for TASWORD TWO...enter TASWORD TWO+! Tasword Two+ adds a few new features, and improves upon some original ones:

1. An expanded and easier-to-understand Main Menu.
2. Easily modify Help Pages from the Menu.
3. Make multiple copies of your document (includes an on-screen copy # indicator).
4. User-definable Paper/Ink colors from the Menu.
5. Easier modification of the left print margin.
6. TS2040 printer is not disabled after use of a full-size printer.

The following program "Taspatch" as listed replaces the original BASIC portion of Tasword Two. Once entered into the computer, the Machine Code (bytes:tasword) portion of your original program needs to be added.

Once both parts are entered, simply go to the Main Menu, and save back a new copy of TASWORD TWO+.

Also included with this article is the supplemental documentation that explains the use of the new features of TASWORD TWO+.

Good luck with your copy of TASWORD TWO+. If you have any other modifications for this program, please send them to me. I also have cassette copies of the "taspatch" BASIC and "tasdoc" text file available for $9.95 postage paid. Bill Ferrebee, Mountaineer Software, 749 Hill Street #6, Parkersburg, WV 26104.

```
Tasword Two+
© Tasman Software 1983
© Mountaineer Software  1986

PRINT text file.................p
SAVE text file.................s
LOAD text file.................l
MERGE text file................m
RETURN to text file............y
define GRAPHICS/printer........g
SAVE Tasword Two+..............t
into BASIC.....................b
EDIT help page(s)..............e
```

TASWORD TWO+ Supplemental Documentation
1986 Mountaineer Software

Welcome to TASWORD TWO+! This program gives you even more Word Processing power, and will make use even easier.

This extra documentation will guide you through the changes made by TASWORD TWO+, and will help you to make the best use of those changes.

MAIN MENU

The Main Menu has been augmented with new choices, and streamlined for easier use. Its use is the same as before: press the corresponding letter to your choice, and ENTER to verify.

PRINT MENU

The PRINT text file Menu has a few new options added to it. First, you may select the number of copies you want printed by typing that number in at the prompt.

Remember to either designate a graphics symbol for form-feed, or space the proper number of spaces between pages, or the text will be printed continously, with no break between copies.

Also, the left margin of your printout may be specified from this menu. Originally, this selection was in the Printer Interface control code selection routine.

PAPER/INK COLOR SELECTION

You may now specify what PAPER/INK color combination you would like to use with TASWORD TWO+. To make your choice, first go to the Main Menu and select "define GRAPHICS/printer....g". When you are asked to "Reset interface/printer codes?", press (y)es. Press ENTER until you get to "Paper/Ink control code=", and use the chart below to enter your choice.

(Remember, some color combinations may not be very suitable to use. This will depend on your TV/Monitor selection.)

PAPER:	BLACK	BLUE	RED	MAGENTA
INK :	Blue.....1	Black.....8	Black.....16	Black.....24
	Red.....2	Red.....10	Blue.....17	Blue.....25
	Magenta....3	Magenta...11	Magenta...19	Red.....26
	Green.....4	Green.....12	Green.....20	Green.....28
	Cyan.....5	Cyan.....13	Cyan.....21	Cyan.....29
	Yellow.....6	Yellow....14	Yellow....22	Yellow....30
	White.....7	White.....15	White.....23	White.....31

PAPER:	GREEN	CYAN	YELLOW	WHITE
INK :	Black.....32	Black.....40	Black.....48	Black.....56
	Blue.....33	Blue.....41	Blue.....49	Blue.....57
	Red.....34	Red.....42	Red.....50	Red.....58
	Magenta...35	Magenta...43	Magenta...51	Magenta...59
	Cyan.....37	Green.....44	Green.....52	Green.....60
	Yellow....38	Yellow....46	Cyan.....53	Cyan.....61
	White.....39	White.....47	White.....55	Yellow....62

HELP PAGES

Help Pages (Main & Extended) may be customized for your own use by using the (e) command on the Main Menu. Once you have entered the particular page into the text file, make sure to replace it as soon as you have made your changes. Then SAVE a new copy of TASWORD TWO+ with your new Help Pages.

Prog:taspatch

```
  10 LET hf=NOT PI: CLS : LET a=USR VAL "64330": GO TO VAL "20"
  11 FOR i=SGN PI TO VAL "9": BEEP VAL ".2",VAL "3": BEEP VAL ".
3",VAL "2": NEXT i: RETURN
  15 POKE VAL "23609",VAL "2": CLEAR VAL "33279": GO SUB VAL "40
00": PRINT AT VAL "10",VAL "5"; FLASH SGN PI;"Do NOT stop the ta
pe";AT SGN PI,NOT PI: LOAD "tasword+"CODE : CLS : PRINT AT VAL "
10",VAL "6"; FLASH SGN PI;"Stop the tape": GO SUB VAL "11": LET
a=USR VAL "59081": GO TO VAL "10"
  20 CLS : LET a=VAL "64"*INT (a/VAL "64"+VAL "0.99"): IF a=VAL
"0" THEN  GO TO VAL "3000"
```

LEMKE SOFTWARE DEVELOPMENT

TS 2068 presents the TS 2068

CHECKBOOK/BUDGET MASTER

The CHECKBOOK/BUDGET MASTER is a checkbook database program and a powerful home or business budget analysis program in one compact/lightning fast program. CBBM was written in BASIC and compiled with the TIMACHINE (by Novelsoft) Basic Compiler!

Program Specification:

Database holds 800 checks (withdrawl or deposits)
Check Recall/Review by: Check Number, Month, Paid To
Functions: REVIEW, LPRINT, DELETE, SUBTOTAL (+ or -)

Budget Analysis: 14 User Defined Categories + 2 (each category has 4 aliases)

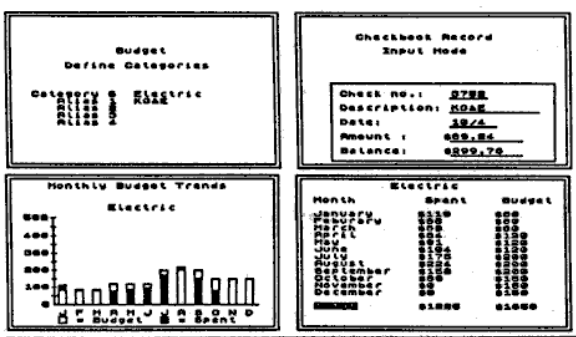
Assign $ Budget to each category (for each month)

Automatically tallies EXPENDITURES as checks are filed and INCOME as deposits are filed!

Update feature: change categories at any time, use the UPDATE command to re-catalog checkbook files

OUTPUT: Tabular or Graphic format
2040 printer or easy full size printer mod.

Get the CHECKBOOK/BUDGET MASTER program from LEMKE SOFTWARE DEVELOPMENT (home of the PIXEL SKETCH and GRAPHICS EDITOR v2.0 the only graphics program to cross the ADVANCED VIDEO MODE barrier providing 32 column, 64 column, and extended color graphics!). CHECKBOOK/BUDGET MASTER and the PIXEL SKETCH and GRAPHICS EDITOR v2.0 are each $19.95 ppd., get yours TODAY!



HERB BOWERS PROUDLY PRESENTS

MONEY MACHINE II

Based on a Popular TV Game Show

STARRING LOVELY MISS BANNA BRITE

The Ultimate Word Game

The all new MONEY MACHINE II FEATURING:

New theme!!

Fantastic Family Fun!!

"BANNA" turning the letters

Hundreds of NEW Puzzles

Solo or 2 or 3 Players

All this for only

T/S 2068 Cassette $12.00 (US)

Includes Postage-via 1st Class

HERB BOWERS, Sr.
2588 Woodshire Circle
Chesapeake, VA 23323
Phone: 804 487-5924

ABVF SOFT

Accurate Business, Bookkeeping and Accounting Software


```
  25 GO SUB VAL "4000": POKE VAL "26703",NOT PI: POKE VAL "26704
",VAL "5":  PRINT AT VAL "4",NOT PI;"PRINT text file.....
....p"
  28 PRINT : PRINT "SAVE text file.................s": PRINT : P
RINT "LOAD text file.................j": PRINT : PRINT "MERGE te
xt file.................m": PRINT : PRINT "RETURN to text file...
.........y"
  30 PRINT : PRINT "define GRAPHICS/printer........g": PRINT : P
RINT "SAVE Tasword Two+.............t": PRINT : PRINT "into BAS
IC.....................b": PRINT : PRINT "EDIT help page(s).....
.........e"
  70 PRINT #SGN PI;AT NOT PI,NOT PI;"          press key
    "
  80 LET a$=INKEY$: IF a$="" THEN  GO TO VAL "80"
  90 LET i=NOT PI: LET b=CODE a$: IF b<VAL "97" THEN  LET b=b+VA
L "32"
 110 IF b=VAL "115" THEN  LET i=VAL "6"
 120 IF b=VAL "106" THEN  LET i=VAL "8"
 125 IF b=VAL "116" THEN  LET i=VAL "16"
 130 IF b=VAL "112" THEN  LET i=VAL "4"
 140 IF b=VAL "121" THEN  LET i=VAL "12"
 150 IF b=VAL "109" THEN  LET i=VAL "10"
 160 IF b=VAL "103" THEN  LET i=VAL "14"
 170 IF b=VAL "98" THEN  LET i=VAL "18"
 175 IF b=VAL "101" THEN  LET i=VAL "20"
 180 IF i>NOT PI THEN  PRINT AT i,VAL "31"; FLASH SGN PI;CHR$ b;
: GO TO VAL "500"
 190 GO TO VAL "80"
 200 CLS : GO SUB VAL "4000": PRINT AT VAL "4",VAL "8";"PRINT OP
TIONS": PRINT "  just press ENTER for default    values given in
 brackets"
 210 LET i=VAL "8": LET j0=VAL "23": PRINT AT i,NOT PI;"Line spa
cing? (1)": GO SUB VAL "6000": IF a$="" THEN  LET a$="1"
 215 POKE VAL "62235",VAL a$
 220 LET i=VAL "10": PRINT AT i,NOT PI;"Start at line? (1)": GO
SUB VAL "6000": IF a$="" THEN  LET a$="1"
 230 LET c=VAL "64"*(INT VAL a$-SGN PI): LET b=c+FN p(VAL "62216
"): LET x=VAL "60045": GO SUB VAL "950"
 240 LET i=VAL "12": PRINT AT i,NOT PI;"Finish at line? (last)":
 GO SUB VAL "6000": IF a$="" THEN  LET b=a-c: GO TO VAL "250"
 245 LET b=VAL "64"*INT VAL a$-c
 250 RANDOMIZE USR VAL "59806": RANDOMIZE USR (FN p(VAL "62472")
)
```

```
 251 LET i=VAL "14": PRINT AT i,NOT PI;"No. of copies (1)": GO S
UB VAL "6000": IF a$="" THEN  LET a$="1"
 252 LET k=VAL a$
 253 LET i=VAL "16": PRINT AT i,NOT PI;"Left margin (";PEEK VAL
"60927";")": GO SUB VAL "6000": IF a$<>"" THEN  POKE VAL "60927"
,VAL a$
 260 CLS : PRINT AT VAL "20",NOT PI;"Press the q key to quit pri
nting"
 270 LET x=VAL "60049": GO SUB VAL "950"
 275 LET c=PEEK VAL "62470": IF c<>NOT PI THEN  LPRINT CHR$ c
 279 FOR i=SGN PI TO k: PRINT AT NOT PI,NOT PI;"Printing Copy #
";i;" of ";k
 280 RANDOMIZE USR VAL "60038"
 281 NEXT i
 285 LET c=PEEK VAL "62471": IF c<>NOT PI THEN  LPRINT CHR$ c
 290 RANDOMIZE USR VAL "59806": POKE VAL "26703",NOT PI: POKE VA
L "26704",VAL "5": GO TO VAL "10"
 300 CLS : GO SUB VAL "4000": PRINT "Printer control graphics ch
ars:"
 305 LET b=VAL "4": FOR i=NOT PI TO VAL "15": LET b=ABS (b-VAL "
4"): PRINT AT i+VAL "4",b;i+VAL "128";CHR$ (i+VAL "128")
 307 FOR j=NOT PI TO VAL "3": LET c=PEEK (VAL "60860"+i*VAL "4"+
j): LET a$=STR$ c: IF c=VAL "255" THEN  LET a$=""
 308 PRINT AT i+VAL "4",VAL "10"+VAL "4"*j;a$: NEXT j: NEXT i
 320 INPUT "Type graphics character code 128-143 (ENTER if finis
hed)";a$
 325 IF a$="" THEN  GO TO VAL "400"
 340 LET b=VAL a$: IF b<VAL "128" OR b>VAL "143" THEN  GO TO VAL
 "320"
 350 PRINT AT VAL "21",VAL "3"; FLASH SGN PI;b; FLASH NOT PI;" "
;CHR$ b
 355  FOR j=NOT PI TO VAL "3": POKE (VAL "60348"+b*VAL "4"+j),VA
L "255": NEXT j
 360 FOR j=NOT PI TO VAL "3": INPUT "Code? (ENTER if finished)";
a$: IF a$="" THEN  GO TO VAL "300"
 370 POKE (VAL "60348"+b*VAL "4"+j),VAL a$: PRINT AT VAL "21",VA
L "10"+VAL "4"*j;VAL a$: NEXT j: GO TO VAL "300"
 400 CLS : GO SUB VAL "4000": PRINT AT VAL "3",NOT PI;"Reset int
erface/printer codes?": LET i=VAL "5": GO SUB VAL "920": CLS : I
F i=NOT PI THEN  GO TO VAL "25"
 401 GO SUB VAL "4000": PRINT AT VAL "3",NOT PI;"just ENTER to k
eep values given:"
 403 LET j0=VAL "27": LET i=VAL "8": LET j=NOT PI: LET a$="Inter
face control code1=": LET x=VAL "60924": GO SUB VAL "850"
 404 LET i=VAL "9": LET j=VAL "18": LET a$="code2=": LET x=VAL "
62470": GO SUB VAL "850"
 405 LET i=VAL "10": LET j=VAL "18": LET a$="code3=": LET x=VAL
"62471": GO SUB VAL "850"
 406 LET i=VAL "11": PRINT AT i,VAL "18";"code4=";FN p(VAL "6247
2"): LET i=VAL "12": LET j0=VAL "24": GO SUB VAL "6000": LET j0=
VAL "27": IF a$<>"" THEN  LET b=VAL a$: LET x=VAL "62472": GO SU
B VAL "950"
 410 LET i=VAL "14": LET j=NOT PI: LET a$="Printer carriage retu
rn=": LET x=VAL "60925": GO SUB VAL "850"
 420 LET i=VAL "16": LET j=NOT PI: LET a$="Printer linefeed=": L
ET x=VAL "60926": GO SUB VAL "850"
 430 LET i=VAL "18": LET x=VAL "58513": PRINT AT i,NOT PI;"Paper
/Ink control code=";PEEK x: GO SUB VAL "6000": IF a$<>"" THEN  P
OKE VAL "58512",VAL "54": POKE x,VAL a$: POKE VAL "58521",VAL "5
4": POKE VAL "58522",VAL a$
 490 GO TO VAL "20"
 500 PRINT #SGN PI;AT NOT PI,NOT PI;" press the "; FLASH SGN PI;
"ENTER"; FLASH NOT PI;" key to proceed";AT SGN PI,NOT PI;"  pres
s "; FLASH SGN PI;"c"; FLASH NOT PI;" to change the choice  "
 510 LET a$=INKEY$: IF a$="c" OR a$="C" THEN  GO TO VAL "20"
 520 IF CODE a$=VAL "13" THEN  GO TO VAL "600"
 530 GO TO VAL "510"
 600 IF b=VAL "116" THEN  GO TO VAL "700"
 610 IF b=VAL "121" THEN  CLS : GO TO VAL "10"
 620 IF b=VAL "115" THEN  CLS : GO TO VAL "1000"
 630 IF b=VAL "109" THEN  GO TO VAL "2000"
 640 IF b=VAL "106" THEN  LET a=USR VAL "59081": LET a=NOT PI: G
O TO VAL "2000"
 650 IF b=VAL "112" THEN  GO TO VAL "200"
 660 IF b=VAL "103" THEN  GO TO VAL "300"
 670 IF b=VAL "101" THEN  CLS : GO TO VAL "5000"
 699 CLS : STOP
 700 CLS : LET i=VAL "8": GO SUB VAL "800": LET a$="tasword+": S
AVE a$ LINE VAL "15"
 710 SAVE a$CODE VAL "54784",VAL "10751": GO SUB VAL "900"
 770 PRINT AT VAL "19",NOT PI;
 780 VERIFY a$: PRINT AT VAL "21",NOT PI;"tasword: basic O.K.;";
AT VAL "19",NOT PI;
 790 VERIFY a$CODE VAL "54784",VAL "10751": PRINT AT VAL "21",VA
L "20";" m/code O.K.": GO TO VAL "25"
 800 PRINT AT i,NOT PI;"Remove ear plug from cassette.": RETURN
```

```
 850 PRINT AT i,j;a$;PEEK x: GO SUB VAL "6000": IF a$<>"" THEN
POKE x,VAL a$
 860 RETURN
 900 PRINT AT VAL "8",NOT PI;"Reconnect ear plug, and rewind";AT
 VAL "10",NOT PI;"and play the tape to verify";AT VAL "12",NOT P
I;a$: RETURN
 920 PRINT AT i,VAL "4";"press  y  for yes";AT i+VAL "2",VAL "11
";"n  for no"
 930 IF INKEY$="n" OR INKEY$="N" THEN  LET i=NOT PI: RETURN
 940 IF INKEY$="y" OR INKEY$="Y" THEN  LET i=SGN PI: RETURN
 945 GO TO VAL "930"
 950 POKE x,b-VAL "256"*INT (b/VAL "256"): POKE (x+SGN PI),INT (
b/VAL "256"): RETURN
1000 LET b=FN p(VAL "62216"): CLS
1005 PRINT AT VAL "8",NOT PI;"Name of text file for saving?": LE
T i=VAL "10": LET j0=NOT PI: GO SUB VAL "6000"
1010 IF LEN a$>VAL "10" THEN  CLS : PRINT AT VAL "12",NOT PI;"to
o many characters - max is 10": GO TO VAL "1005"
1020 IF LEN a$=NOT PI THEN : CLS : PRINT AT VAL "12",NOT PI;"the
re must be a name": GO TO VAL "1005"
1030 LET i=VAL "12": GO SUB VAL "800": SAVE a$CODE b,a: CLS
1040 PRINT AT VAL "8",NOT PI;"text file ";a$;" saved:";AT VAL "1
0",NOT PI;a;" bytes,",a/PEEK VAL "62237";" lines"
1050 PRINT AT VAL "12",NOT PI;"do you want to verify ";a$;" ?":
LET i=VAL "14": GO SUB VAL "920": IF i=NOT PI THEN  GO TO VAL "2
0"
1100 CLS : GO SUB VAL "900": VERIFY a$CODE b,a
1110 PRINT AT VAL "21",VAL "6";"text file verified": GO TO VAL "
25"
2000 CLS : PRINT AT VAL "8",NOT PI;"type the name of the text fi
le";AT VAL "10",NOT PI;"and press ENTER"
2010 PRINT AT VAL "12",VAL "2";"just press ENTER to load the";AT
 VAL "14",VAL "2";"first text file on the tape"
2020 LET j0=NOT PI: LET i=VAL "16": GO SUB VAL "6000": PRINT AT
VAL "18",VAL "9";"Play the tape"
2030 LET b=FN p(VAL "62216"): LOAD a$CODE (a+b),((FN p(VAL "6222
1")+VAL "22")*VAL "64"-a): GO TO VAL "10"
3000 FOR i=VAL "23296" TO VAL "23361": POKE i,VAL "32": NEXT i
3005 POKE VAL "23362",NOT PI
3010 PRINT AT VAL "8",NOT PI;"type word to be replaced / found"
3012 LET j0=NOT PI: LET i=VAL "10": GO SUB VAL "6000": IF a$=""
THEN  GO TO VAL "10"
3020 LET j=NOT PI: FOR i=SGN PI TO LEN a$: POKE VAL "23297"+i,CO
DE a$(i): IF a$(i)=" " THEN  LET j=j+SGN PI
3021 NEXT i
3022 IF j<>NOT PI THEN  CLS : PRINT AT VAL "12",NOT PI;"just a w
ord - no spaces allowed": GO TO VAL "3000"
3025 POKE VAL "23297",LEN a$
3030 PRINT AT VAL "12",NOT PI;"with (just ENTER for find only)":
 LET i=VAL "14": GO SUB VAL "6000"
3040 IF a$="" THEN  POKE VAL "23362",SGN PI: GO TO VAL "3060"
3050 FOR i=SGN PI TO LEN a$: POKE VAL "23329"+i,CODE a$(i): NEXT
 i
3060 LET a=USR VAL "64955": LET a=USR VAL "64333": GO TO VAL "20
"
4000 PRINT AT NOT PI,VAL "10";"Tasword Two";AT SGN PI,VAL "5";"
 Tasman Software 1983";AT VAL "2",VAL "2";" Mountaineer Softwa
re  1986": RETURN
5000 GO SUB VAL "4000": PRINT AT VAL "4",VAL "6";"EDIT help page
(s)";AT VAL "7",VAL "4";"OPEN :  1Main  2Extended";TAB VAL "4";"
CLOSE:  3Main  4Extended"
5010 LET a$=INKEY$: IF a$="" THEN  GO TO VAL "5010"
5020 IF VAL a$<SGN PI OR VAL a$>VAL "4" THEN  GO TO VAL "5010"
5030 IF a$="2" OR a$="4" THEN  LET hf=SGN PI
5040 PRINT AT VAL "10",VAL "12"; FLASH SGN PI;"WORKING!"
5050 LET c=VAL "54784": IF hf=SGN PI THEN  LET c=VAL "56320"
5060 FOR i=NOT PI TO VAL "1536"
5070 IF a$="1" OR a$="2" THEN  POKE (VAL "33280"+i),PEEK (c+i)
5075 IF a$="3" OR a$="4" THEN  POKE (c+i),PEEK (VAL "33280"+i)
5080 NEXT i: LET hf=NOT PI: GO TO VAL "20"
6000 LET a$="": PRINT AT i,j0; FLASH SGN PI;" "
6010 LET j=j0: IF INKEY$<>"" THEN  GO TO VAL "6010"
6020 LET b$=INKEY$
6030 IF b$="" THEN  GO TO VAL "6020"
6040 IF CODE b$=VAL "13" THEN  PRINT AT i,j;" ": RETURN
6050 IF CODE b$<>VAL "12" THEN  GO TO VAL "6170"
6060 IF j=j0 THEN  GO TO VAL "6200"
6070 LET j=j-SGN PI: PRINT AT i,j; FLASH SGN PI;" "; FLASH NOT P
I;" ": LET a$=a$( TO j-j0): GO TO VAL "6200"
6170 IF CODE b$<VAL "32" OR CODE b$>VAL "127" THEN  GO TO VAL "6
200"
6180 BEEP VAL ".005",VAL "5": PRINT AT i,j;b$; FLASH SGN PI;" ":
 LET j=j+SGN PI: LET a$=a$+b$
6190 IF j=VAL "32" THEN  PRINT AT i+SGN PI,NOT PI;" ": RETURN
6200 IF INKEY$<>"" THEN  GO TO VAL "6200"
6210 GO TO VAL "6020"
7000 DEF FN p(x)=PEEK x+VAL "256"*PEEK (x+SGN PI)
```

SOFTWARE

Oliger 2.1 Disk System Update

Reviewed By Dick Wagner

The John Oliger Company is now delivering their V2.1 EPROM for the SAFE disk system. There have been quite a few reviews of the original V1.1 DOS including the mother board, board "A" and "B", and NMI (non-maskable interupt) switch add-on used for state-of-computer saves. There has been an improvement in the NMI switch part of board "B" and it is now part of the unit. A disable switch is also added to this board. I added this switch because of a problem experienced in making cassette saves when an A&J I/F was in place. This switch solved it. This review is intended to bring information to the reader about the new V2.1 DOS EPROM commands.

The new EPROM V2.1 is available for $20 ppd. or the user can send in the V1.a EPROM for $15 ppd. and get it re-programmed.

I must say that John has not let the V1.1 user down in the least as it is possible to use ALL of those old files. We can continue to use the old commands for loading old files, but new commands are required to SAVE to new FORMAT and to LOAD those saves. An old disk cannot be saved on by this new system without proper formatting.

The Following list of commands shows the degree of file storage capability now built into SAFE DOS. It is nice not to have to learn a whole new series of strange commands as most of these are duplicate in action to the cassette commands.

```
LET /S=n                        LET /T=n
LET /D=n                        LET /H=n
LET /P=D & LET /P=T             LET /P=D/B & LET /P=T/B
FORMAT /"FILENAME"              SAVE /"FILENAME"
SAVE /"FILENAME" LINE n         LOAD /"FILENAME"
SAVE/@                          LOAD or LOAD/@
SAVE /"FILENAME" CODE n,n       LOAD /"FILENAME" CODE n,n
LOAD /"FILENAME" CODE n         LOAD /"FILENAME" CODE
SAVE /"FILENAME" SCREEN$        LOAD /"FILENAME" SCREEN$
SAVE /"FILENAME" DATA X()       LOAD /"FILENAME" DATA X()
SAVE /"FILENAME" DATA X$()      LOAD /"FILENAME" DATA X$
SAVE /"FILENAME" VAL            LOAD /"FILENAME" VAL
SAVE /"FILENAME" ABS            LOAD /"FILENAME" ABS
LOAD /n                         CAT
MOVE /                          COPY/
NMI (pushbutton save)   RESTORE /"OLDNAME" TO "NEWNAME"
```

Several JLO commands have been added by using keywords. For instance, the VAL command saves and loads the BASIC program variables only. Thus, those variables are not cleared on loading. Using ABS in the command, saves everything as the old SAVE /n or NMI save does. These files are retrieved by LOAD /"n" ABS where "n" is the number selected for the save ID.

CAT is the catalog system used by V2.1 and is simple to use. The user can call up CAT on a disk with a program in the computer without loss. CAT displays the disk name (it must have a name, even " "). FORMATTED shows the number of tracks, the total capacity in cylinders, and bytes. FREE shows the number of cylinders left. Cylinders are 5k blocks of storage. Also the display shows the file name, the basis the storage was made on (STATE, BASIC, BYTES), the number of cylinders used and the number of bytes in each file.

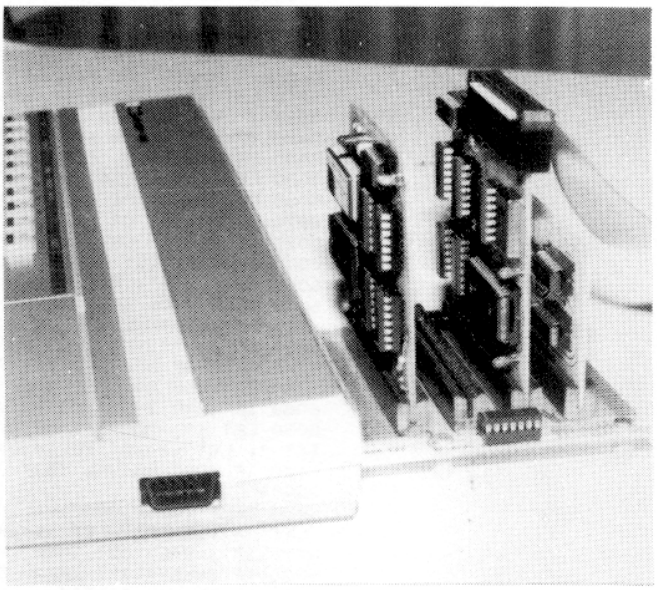
Some interesting operations are possible. File names can be changed so NMI saves by number can be changed to words later. NMI saves are memory hungry (at least 49664 bytes). After a NMI save has been made the file can be LOADed and then saved again as DATA, VAL, SCREEN$, and CODE desired. This may or may not save cylinders of space, depending upon program length. Overwriting is interesting as the first time a name is used, the space in cylinders for that name is established. Overwriting with a changed program may require more space than first established. If so, you get a report "FILE TOO LARGE". The old file is not lost nor do you loose the new program. This requires a save under a new name and if the disk lacks enough space a "DISK FULL" report is shown. If a program is to be saved with an identical name of a program already on that disk a TOOT is given as a warning, and there is 5 seconds to cancel the instructions with ENTER.

SAVE/0 is still available and can be used to show information not shown in CAT, such as 2068 or Spectrum systems (works with both), special disk use such as TASWORD, MSCRIPT, arcade games, utilities, ect. This file will hold 1 1/2k of information.

This system is a pleasure to use and it overcomes the major complaint about V1.1 system-limited number of files. I have no experience with other DOS, so I can't really make a comparison this way. John entered into the development of V2 after extensive delays ocurred in obtaining the Ray Kingsley DOS. Ray has a preliminary DOS operating and rumor has it, that the final system will not be found wanting in features available to other makes of computers. I was in hopes of obtaining Ray Kingsley's system but felt that John's SAFE improvements would do what I wanted with the drives I am now using.

One improvement I would like to see, is an ERASE command to delete a file from disk. I tried saving an empty file with an identical name/save basis. The file appeared to be deleted, but the number of bytes was not changed. Even an empty file must have an identical save basis if this method is use.

For further information on the Oliger SAFE V2.1, write to: The John Oliger Company, 11601 Whidbey Dr., Cumberland, IN 46229. Include a legal size S.A.S.E.



I.S.T.U.G. TS2068 BBS

Reviewed By Ralph McCrum

I am sure that every BBS "addict" has promised with each months growing phone bill, that they will cut down on "calling"...only to do the same thing again the following month...or wished for his/her BBS, so that other "addicts" would call him/her instead! But, not having disc drives, or money for high-priced software, owning a BBS seemed impossible.

Members of the Indiana Sinclair Timex User Group felt the need for a dependable, low-cost BBS system, and went to work. Now if you own an "unexpanded" Timex Sinclair 2068, a TS2050 modem, and a TS2040 printer, you can operate your own BBS.

The I.S.T.U.G. TS2068 BBS program loads in about 48 seconds from cassette tape, and uses just 8.5k RAM of memory. It functions mainly in simple BASIC, with the codeing hidden neatly in line 0. An easy to use SYSOP editor, lets you establish the time and date. You can quick scan the message base, Read, leave, or delete messages, Load or Save the message base. You can return to the SYSOP editor from anyplace in the program by simply pushing [symbol/shift] STEP. A real time clock keeps the time and date, and even advances the date at midnight. The message base will hold 40 messages--535 characters long. Each message is stamped with the time and date of when it was left. There is room for one to four screens of bulletins. The system automatically prints to the 2040 printer, the name of the caller, what messages they read (or leave), along with a "page" for the SYSOP, and the elapsed time that the caller was on-line. All of this makes it easy to keep records, and manage your BBS.

The BBS callers have a quick scan option that only lists who the message is for, and who it is from, and the time/date of the messages. If the caller is in a hurry, he can abort the scan and return to the menu. There is a check time option, a "chat with the SYSOP" option, and a "read users log" option. And of course, a read/leave message option.

The I.S.T.U.G. BBS is easy to operate, and "crash proof"...and most important.... dependable...providing many hours of just plain fun. It comes with six pages of easy-to-understand documentation, that takes you step-by-step into big time BBSing.

The work of the software's programmer Paul Holmgren (and BBS SYSOP, Willie Jones) can be viewed by calling the I.S.T.U.G. board at 1-317-898-3903 (24 hours). Modem settings: 7-1-E OR, you can get your own copy of the BBS program, by sending $16 (postage paid) U.S. funds to: Willie Jones, 10126 E. 33rd Street, Indianapolis, IN 46236. When you order, you might mention to Willie that you wouldn't mind recieving his "Burn graphics one-time command" utility also...just to make it a little more interesting.

COLOSSUS

Reviewed By Duncan Teague

Like it's namesake, the Colossus of Rhodes, Lemke Software Development's graphic banner designer utility "Colossus" stands head and shoulders above similar software. Colossus contains features that heavyweight software-developer Broderbund Software has just recently added to it's popular program "The Print Shop".

Because of the popularity of "The Print Shop", a graphic utility for lesser computers than our beloved TS2068, everyone knows what a "banner" is. "The Print Shop" allows any of several type fonts to be used for the message portion of the banner. It also permits a graphic picture to be included in front of and or behind the message.

What makes banners produced by "Colossus" different from those made by "The Print Shop" is "Colossus'" ability to include more than one type font in a single banner and the option of placing graphic pictures anywhere within the banner.

There are, unfortunately, no graphic pictures included with the program. The user must draw his or her own. Fortunately, the program contains an option for using redefined keys to print graphic symbols on the screen.

There are twenty symbol keys, nineteen stored graphics plus a blank space. Fourteen redefined keys duplicate the shifted and unshifted graphic symbols on the 1-5 and 7-8 keys. The other six symbols include a right triangle in four different orientations, a hollow square, and a small, centered square dot.

The text can be typed into the banner in any of three sizes and in two different type fonts. One font is the standard TS2068 system font; and the other is a calligraphy font called "Chancery". Either of these two fonts can be modified to give a different look. The modifying options are bold, modern, and italics.

The effects of the modifiers are additive. One can, for example, use bold-modern-chancery font or a modern-italic-system font. All possible combinations of modified and unmodified fonts would give the user a choice of sixteen different type styles to be used in any one banner.

"Colossus" can store up to 1024 columns, or 32 screens, of data. It can horizontally scroll one column at a time while you are working on the banner. Rows can also be inserted or deleted. A portion of the banner can be copied to another portion. In a similar manner segments of the banner can be erased.

Once the banner has been completed, you can view your creation sequentially. The banner is scrolled from right to left across the screen a selectable number of columns at a time. A special feature allows you to set the scroll rate at thirty-two columns, one full screen, every four seconds. If successive screens have been designed correctly, your banner is animated.

The banner can be printed horizontally on a TS2040 printer or on a full size printer. The only catch to using a full size printer is that it must be supported by Zebra Systems Inc.'s "ZPrint-80" print driver. My C.Itoh Prowriter is not. Gr-r-r! Along with printing, "Colossus" Saves and Loads banners via cassette tape.

A welcome piece of support software which would complement "Colossus" and the previously reviewed "Pixel Sketch and Graphics Editor" would be a utility for designing your own type fonts. I understand that one will be published in the upcoming issue of TIME DESIGNS. I'm looking forward to it.

"Colossus" is $19.95 and is available from Lemke Software Development, 2144 White Oak, Wichita, KS 67207. The program comes with two samples: "LSD-banner" and "Animation".

ACZ General Ledger

Reviewed By Dennis Silvestri

ACZ GENERAL LEDGER 2.000 is a small business accounting system for the TS 2068 computer. This program will provide the small business with the following financial reports: Monthly and Year To Date Income Statements, Ledger Detail, Balance Sheets, Chart of Accounts, Trail Balance, and Journal Entries. This double entry system has a maximum capacity of 800 entries and accounts per month in any combinantion and a maximum of 150 named accounts. The program will calculate sums of up to $999,999.00. Printing of all transactions is to the TS 2040 printer only. You can also use the optional Cardco brand numeric keypad. This keypad plugs into the joystick port and is used to speed entry of numerical data. I am not familiar with this keypad and did not have one available for review, so I am unable to say how much if any this keypad will increase the entry of data.

There are two parts to this program. The system "set-up" and the Transaction Entry. The system set-up determines the format of the business financial statements as well as the position of the accounts in all reports and establishes Account Numbers. This set-up routine also requires the user to make and save three data programs, that must be used with the transaction entry program. Helpful menu's and prompts guide you along the way, while creating the set-up routine.

When the system set-up is complete you are ready to Load in the transaction entry program and the tapes that were created from the set-up. Not all three set-up tapes are loaded together. What determines which of these tapes are used depends on the following: If you are using the system for the first time, starting a new month, or adding additional transactions to the current month. It is with this transactions entry that all General Ledger Accounting is done including printing of Financial Statements and Saving of all records. Again, helpful menu's and prompts guide you along the way.

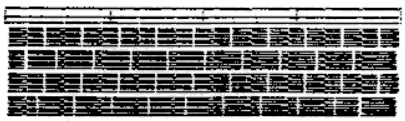
Since this program is only available on tape, there is time lost due to saving, verifying, and merging of the various routines. However, this is minor when compared to the time it would take to format and enter all transactions without the use of a computer.

ACZ GENERAL LEDGER comes with extensive documentation that can be read and understood, providing the user takes the necessary time to thoroughly read it. Included in the documentation are financial records for a fictitious business. Before converting your books to this system the user should work with the given example.

ACZ GENERAL LEDGER 2.000 is available from WMJ Data Systems, 4 Butterfly Dr., Hauppauge, NY 11788. Cost is $19.95 plus $3 for S&H.

BRICKWORKER

Reviewed By R. Lussier



BRICKWORKER is a utility-type of program for those who require help, or are interested in the "art" of brickworking. BRICKWORKER will operate on either the Spectrum or the un-aided TS 2068 computer.

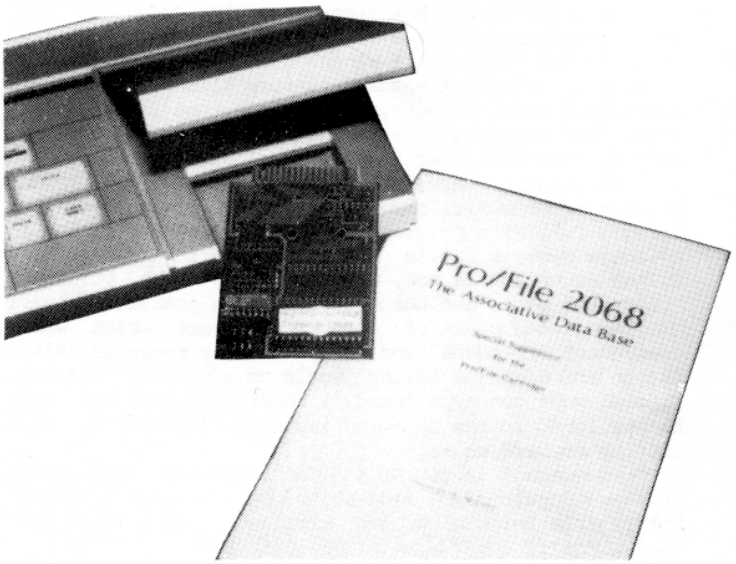
The program contains illustrations of Bonding, Foundations, Brickwork Calculator, Mortar mixes, and Decorative finishes. The screen illustrations are good and very useful. The program itself is written in BASIC and is very User friendly...an overall good quality program.

Another program that the same software company has on hand is called PAVEMASTER which deals with the design and quantity calculation for constructing Patios and Driveways, ect., using various sizes of paving blocks and slabs.

If interested in such programs, then contact-Konkrete Software, 6 Willowcourt Ave., Kenton, Harrow, Middlesex HA3 8ES, Great Britain.

PRO/FILE Cartridge

Reviewed By Robert Fischer



Probably the most useful type of computer program is the data base since almost anyone can find a need for it and because it can be effectively used without a printer.

For Timex computers, Tom Woods has twice provided high quality programs in this field with ZX PRO/FILE for the TS 1000 and then with PRO/FILE 2068. Just get out old copies of Timex oriented magazines and newsletters and you will find consistently high praise for these programs.

Now Tom has a new version on cartridge for the TS 2068. This is much more than you may expect. Sure, it loads instantly and has much greater capacity (37,000 bytes), but it also has many new capabilities.

First, let me briefly list the elements of the program which are essentially unchanged. Each file can have up to 15 lines of 32 characters each. Multi-word searches are still available and the program is still very flexible, easy to use, and very fast. Besides these things, the improvements found in the manual which could be added yourself are almost all included as is Tom's machine code sort routine and a DATA save (thank goodness).

About the only negative changes are an extremely small reduction in speed due to extensive bank switching and a reduced ability to make changes due to the use of a cartridge, although you can make some.

Enhancements to the original routines include the ability to calculate numeric information on any line you wish (this refers to the TALLY routine in the original manual), built in programming allowing a choice of seven printer interfaces, the ability (when using a full size printer) to print different file lines on the same line of paper, TALLY information can be printed out in the position you wish as well, and when adding more than one file at a time, you can bypass the main menu.

Totally new routines include duplication of files (saves time with files that are similar), separate sets of files can be merged together, you can start a search from any point in the files (useful when your printer breaks dwon halfway through a search), and you can adjust the INK and PAPER for color or black and white TVs. For those who wish to use an IBM compatible keyboard, you'll be pleased to know that the cartridge includes the necessary software. His documentation shows you what hardware modifications are needed.

For those with special drive systems, you can make the program exit to regular BASIC to permit saving on these devices (instructions for a couple are provided as examples). Of course, you can also write special BASIC routines for other purposes as well.

All the above improvements are valuable and worth the investment, especially if you don't have the original PRO/FILE 2068, but I saved the best for last. When you input a search command, besides being able to search for single or multiple words you can also perform a NOT search or an OR search! In fact, AND, OR, and NOT can all be combined! The power this provides is fantastic and as you use it, you'll wonder how you ever worked without it.

For all of the following examples, I'll use situations I actually deal with in the band classes I teach. The AND search is pretty straight forward. If I search for "Flute AND 12th-" I would find only the flute players who are seniors. If I extend it to "Flute AND 12th- AND MB-" I would obtain all senior flute players in the marching band (MB- is my code for marching bank). That's pretty easy to understand.

The NOT search is essentially the opposite of AND. To compare with the above example, if I search for "Flute NOT 12th- NOT MB-" I would get every flute player who is not a senior and who is not in marching bank. Since I have instrumentalists who are in the color guard during marching season, a more typical situation would be a need to list only those marching band members who are not in the color gurard. I just do this search: "MB- NOT RIFLE NOT FLAG". It is a little unusual at first searching from this opposite viewpoint, but it soon becomes second nature and Tom explains it well in the documentation.

The OR search is especially useful to me. Often I do a printout of the band according to the section they are in. Therefore I would first search for "Piccolo" and when done, search for "Flute" and then "Clarinet", ect. All those searches were annoying, but with OR I can search for "Piccolo OR Flute OR Clarinent OR Saxaphone" and the computer will print all the piccolo players followed by flutes, ect. I'm sure most of you can think of times you wished you had this capability.

The OR search can get a bit complicated when used in combination with the other commands. Think of it as a divider. For example, the command "Flute NOT 12th- OR Clarinet AND 12th- OR Saxophone" would result in the computer first finding every flute player who IS NOT a senior. When this is done, the computer will find every clarinet player who IS a senior. Finally it will find every saxophone player. See how the use of OR divides up the search command? If not, Tom explains it better than I do and gives examples of what to watch out for.

Simply put, to me the new search capabilities alone are worth the price and everything is just a bonus. I strongly recommend it to anyone who needs to keep track of names, addresses, phone numbers, recipes, collections or just about any other records you can think of.

The PRO/FILE Cartridge for the TS 2068 is priced at $59.95 plus $1.50 for postage, and is available from the author, Thomas B. Woods, PO Box 64, Jefferson, NH 03583.

Beginning Z80 Machine Code

PART SIX

By Syd Wyncoop

Editor: Those readers following along with Syd's Machine Code series, should note that there were two textual errors in the Nov/Dec '86 issue, that need to be corrected. Last paragraph, first column on page 19 should read, "the status of these bits is unimportant." The last sentence of the fourth paragraph on page 20 should read, "the only flags affected are the P/V and zero flags. The P/V flag is reset if BC=0. The zero flag is set to indicate a match, just as in a normal CP instruction."

This lesson we will discuss the stack and the instructions which use the stack. What is a stack? A stack is simply an area of consecutive bytes of memory which are used for storage by the CPU. The CPU cannot operate without a stack. We will find that we too can use the stack, if we are careful.

Remember our earlier discussion of ROM, RAM and boxes? If not, you need some back issues! Think of our CPU's stack as a stack of boxes (memory locations). You can remove or add to the top of the stack easily, but try to remove or add a box somewhere in the middle and the stack topples. CPU's stack works the same way except it grows down from the top as if the boxes were suspended from the ceiling. Therefore, we actually add to the bottom of the stack.

There is a special register inside CPU dedicated to keeping track of the stack. Its mnemonic is SP which means Stack Pointer. SP contains the address of the last location on the stack.

All information on the stack is stored in the usual two byte format used for addresses. We can place information on the stack (PUSH) or remove it from the stack (POP). Our friend CPU automatically adjusts the SP with each operation by the required two bytes. It is important to realize that even though SP is adjusted to point to the correct location (box), the information is still there until it is overwritten. See Fig.1 to make this clearer.

The PUSH and POP instructions can add/remove information to/from the stack and any register pair. For instance, if we wish to stack the contents of the B register, we need to PUSH BC. We will have also stacked the C register, since we must use a register pair.

Last issue we learned the CALL instruction. It uses the stack to save the value of PC in order to know where to return to. In effect, CALL executes a PUSH PC, JP to new location and complete the subroutine, and then a POP PC (Ret) and continue executing the program from the byte after the CALL instruction.

The next instruction is Ld SP,HL. This is a simple assignment instruction. Whatever value held in HL will be copied into SP, not the stack. Remember, most instructions assume all values on the stack to be valid addresses, even if they are data, so it is important to know where SP is.

The last instruction affecting the stack is EX (SP),HL. It will exchange the contents of the address referenced by the SP with the value held in HL. Assume HL + 1040h and SP = 9050h and address 9050h = 59 and address 9051h = 68. After the EX (SP),HL instruction is executed, their new values will be; HL = 6859h, SP = 9050h, address 9050h = 40h and address 9051h = 10h. Notice that SP is unaffected. However, the contents of the last stack entry are changed.

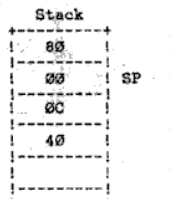
This is a good time to introduce the other exchange instructions. They are all fairly easily understood and are listed in the chart. Note that an exchange merely swaps the contents of the affected registers and no others, neither are any flags affected, except for the EX AF,AF' instruction. The EX AF,AF' exchanges only these registers while the more general EXX exchanges BC, DE & HL with BC', DE' & HL'.

These exchange instructions do not actually change the register contents. Consider the EX AF,AF' instruction. The AF register becomes the AF' register and the previous AF' register pair becomes the new AF register pair. This is important as the contents of registers can be stored out of the way for later retrieval. It also means we must be sure of which set of registers are in use.

The EX DE,HL instruction is very useful and will exchange the contents of DE with the contents of HL. This is the same as if there had been an instruction to Ld DE,HL and Ld HL,DE without disturbing any of the values. A series of PUSHes and POPs would be needed to accomplish the same result. For example, lets EX BC,HL (there is no such instruction):

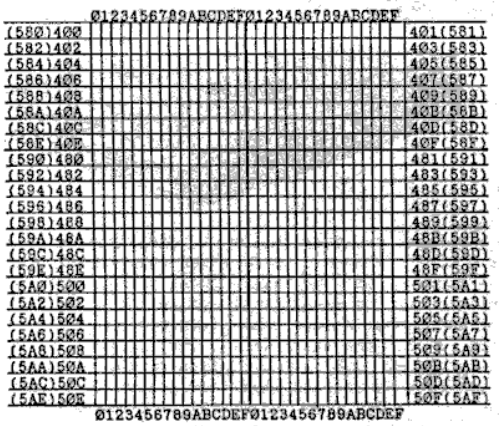
```
PUSH BC
PUSH HL
POP BC
POP HL
```

Figure 1



Assume the top two locations of the stack contain the address 8ØØØh, that the SP is set at 61FEh and HL contains ØC4Øh. A Push HL instruction will load ØC4Ø in the next two stack locations, 61FDh and 61FCh, and Dec SP twice, thereby making SP = 61FCh. A subsequent Pop HL instruction will then Inc SP twice while placing ØC4Øh into HL. It is important that even though SP has been adjusted, locations 61FCh and 61FDh still contain the address ØC4Øh and will until it is overwritten by another stack operation.

Figure 2



Take first 3 digits from nearest edge
Take 4th digit from top/bottom edges
D-File addresses are not in brackets
A-File addresses are in the brackets

Chart 1

Stack	Exchange
Push rr	Ex AF,AF'
Pop rr	Exx
Ld SP,HL	Ex DE,HL
Ld SP,nn	Ex (SP),HL
Ld SP,(nn)	
Inc SP	
Dec SP	

NEW BOOKS FOR THE NEW YEAR!
FROM
RMG ENTERPRISES
****
2068 ROM MANUSCRIPT

By DR. LLOYD DREGER OF SMUG is a FULL DISASSEMBLY of the 2068 ROM, with notes and comments. If you are into MC programming on the 2068, this is one volume that will serve you well! Over 200 pages of information with an index to routines with starting addresses in both HEX and DEC! Also includes a full cross-reference to SPECTRUM ROM!

Our price is only $16.95 plus $2.50 ph.

INTRODUCTION TO 2068 MACHINE CODE PROGRAMMING

Also by DR. LLOYD DREGER OF SMUG (SINCLAIR MILWAUKIE USERS GROUP) is the first of a proposed 2 book series on this subject. This one has A LOT if information that has not been explained too well up to now. Again, over 200 pages of educational material. Written to be used as a self study course in beginning MC, we feel that it is well worth the money! Also only $16.95 plus $2.50 ph.

SPECIAL!!!
BOTH BOOKS-ONLY $30.00
PLUS $3.00 PH!

SEND TO:
RMG ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OR 97045
503/ 655-7484
TUE-SAT 10 AM TO 10 PM

BYTE POWER

ST CLASS MAGAZINE

A major breakthrough for the T/S 2068. BYTE POWER, a highly sophisticated computerized magazine on cassette.

No longer will you have to type in long, fastidious programs...JUST LOAD AND RUN!

BYTE POWER is the ultimate magazine for the T/S 2068. Each tape is full of programs, a real T/S 2068 owner's dream.

BYTE POWER is a software based magazine with over 130 programs per year, most of them in fast Machine Language. Programs such as TALKING CLOCK, EASYword word-processor, GRAND PRIX a 3D race, and SSP (Small Screen Painter) are some examples of the high quality programs published in BYTE POWER Magazine.

BYTE POWER brings you this quality programming at a very small cost, LESS THAN 39 CENTS A PROGRAM, based on 1 year subscription.

Plus you get Software and Hardware reviews and tips to help you design better programs!

1 ISSUE (SAMPLE).....$5.50
6 MONTH SUBSCRIPTION (6 ISSUES).....$29.99
1 YEAR SUBSCRIPTION (12 ISSUES).....$49.99
2 YEAR SUBSCRIPTION (24 ISSUES).....$89.99

All prices in US FUNDS.

SEND CHECK OR MONEY ORDER TO

BYTE POWER
1748 Meadowview Avenue,
Pickering, Ontario, Canada L1V 3G8

Note that the information was moved from the stack into a different register pair than it originated from. This is a very useful tool to have at our disposal. However, we must be aware of what we are doing or we may find ourselves expecting data at a location other than where it ended up.

You will no doubt have noticed that I have rater laboriously explained the many instructions we have learned up to this point. The truth of the matter is that I have been trying to walk the fine line of too much detail/not enough detail. I hope there has been enough to get you started without boring anyone.

We now have enough instructions to begin programming. I firmly believe the only way to learn any language is to use it. With that in mind, we will concentrate more on accomplishing some task and less on the instructions. I must assume that if you are still with me, you have by now acquired some good books to supplement your learning.

We will need to be able to "see" if our programs are completing the task as we desire, therefore, we will initially write programs that will affect the display file. This will necessitate two separate discussions, as the TS1000 and TS2068 each handle their display files differently. You may wish to skip the section which does not pertain to your computer, but I think you will find it beneficial to read.

Before we jump right into it though, we need to look at the Sinclair manual again. Towards the back of the manual you will find a section on the system variables. These are variables used by the Basic operating system to keep track of various items. Many of these will prove useful to us and several others are required to be under our full control. I will use Sinclair's names and explain each one as we need it. You should take a moment to review this section of the manual as we will become comfortable with many of the system variables.

Timex Sinclair 1000

The display file (D-File) is arranged as 24 rows of 33 characters. The last character in each row is an end-of-line (EOL) marker, which is CHR$ 118 (the code for Enter). In addition the very first character is an EOL marker. We must never ever overwrite any of the EOL markers. If we overwrite any of the EOL markers, the system will crash!

This description only applies to a fully expanded system (greater than 3.25k). The D-File is collapsed to 25 EOL markers in a smaller system. I will assume yours is fully expanded.

Since the D-File moves about in memory as your Basic program expands and contracts, its location is held in a system variable known as none other than D-File. This means that we can always locate the D-File with the instruction Ld HL,(D-File).

The easiest way to print to the D-File is to use the RST 10h instruction as that is where Sinclair has placed the print routine. RST 10h will print whatever character is in the A register. Enter the following to get a full screen of asterisks:

Listing 1

```
0E16    Start   Ld C,18h        ;line counter
0620    Loop1   Ld B,20h        ;characters/line counter
3E17    Loop2   Ld A,17h        ;character to print
D7              Rst 10h         ;go print it
10FB            DJNZ,Loop2      ;until line is full
0D              Dec C           ;count one line done
20F6            Jr NZ,Loop1     ;another line?
C9      Done    Ret             ;return to basic
```

While RST 10 is the easy way, it is only a minor improvement over Basic. That's because we are using the same routine as Basic uses. The advantage is that we didn't have to keep track of the EOL markers.

The fastest way to print to the screen is by direct pokes, even from Basic. Enter the following for an almost instant screen fill:

Listing 2

```
2A0C40  Start   Ld HL,(D-File)  ;get D-File location
0E18            Ld C,18h        ;line counter
23      Loop1   Inc HL          ;get past EOL
0620            Ld B,20h        ;characters/line counter
3617    Loop2   Ld (HL),17h     ;poke character onto screen
23              Inc HL          ;advance print position
10FB            DJNZ,Loop2      ;go do again?
0D              Dec C           ;count one line done
20F5            Jr NZ,Loop1     ;do another line?
C9      Done    Ret             ;return to Basic
```

You should have noticed that this method allowed printing on all lines. There is a system variable, DF_SZ, which can be poked from MC or Basic to allow full screen printing, however, the system can be easily crashed if not properly handled. Also, the number 17h can be any printable character code.

Now for an all purpose, generic print routine:

Listing 3

```
1A      Print   Ld A,(DE)       ;check for EOL marker
FE76            Cp 76h
2001            Jr NZ,NoEOL
13              Inc DE          ;get past EOL marker
7E      NoEOL   Ld A,(HL)       ;get character to print
FEFF            Cp FFh          ;check for end of text
C8      Exit    Ret Z           ;and return if reached
12              Ld (DE),A       ;print it
23              Inc HL          ;advance character pointer
13              Inc DE          ;advance print position
18F3            Jr Print        ;do it again
```

The print routine is useless by itself. Upon entry, HL must contain the address of the first character to print and DE must contain the address in the D-File to print at. Enter the following routine to understand how you would set-up HL & DE and Call this routine at Print.

Listing 4

```
2A0C40  Entry   Ld HL,(D-File)  ;get D-File location
EB              EX DE,HL        ;into DE
210041          Ld HL,Text      ;get address of text
                                ;address assumed to be 4100h
CD8240          Call Print      ;go print message
                                ;address assumed to be 4082h
C9      Done    Ret             ;return to basic
```

And now a message must be stored at 4100h. Enter these hex codes to address 4100h:

Hex Dump 1

```
39 2D 2E 38 00 2E 38 00 26 00 39 2A 38 39 1B
00 2E 00 38 3A 37 2A 00 2D 34 35 2A 00 2E 39
00 00 3C 34 37 30 38 1B FF
```

Note that the print routine requires a terminating byte FFh in order to exit. Our test should now work with the command Rand Usr Entry.

Timex Sinclair 2068

The D-File consists of 192 lines of 32 bytes/line for the character information and 24 lines of 32 bytes/line for the attribute information. The last 768 bytes are known as the Attribute File (A-File). The D-File is fixed in memory at address 4000h and the A-File resides at 5800h. We will discuss the A-File at another time, as its purpose is to hold the color attributes of each character square. We will therefore not be using the A-File at this time.

The organization of the D-File is not what you would expect. Each character is eight pixels by eight pixels (one character square). The eight pixels across fit nicely in one byte, hence the 32 bytes across each line. The problem is the eight bytes needed to make each character are not stored consecutively. Looking at Fig.2, you can see that the D-File is split in three sections of 64 lines each. Within each section, the eight lines which comprise each character are 256 bytes apart (8 lines * 32 character spaces). The junction of two sections is where there is a difference as the sequence begins to repeat. Study Fig.1 to make this clear. I am told this unique structure has something to do with the way in which a TV draws its scan lines. Since I understand very little about the hardware, I must claim ignorance and accept this explanation.

This means that the easiest way to print to the screen is by using RST 10h, which is where Sinclair chose to start an all purpose print routine. Once again though, things are not as easy as they would seem. The 2068 uses channels and streams to direct the traffic (we will discuss channels and streams later). This means that we must be sure we know where we are directing the output of RST 10h or else we will have no idea where it will end up.

Do not allow the D-File structure to put you off. We can still write to it if we understand its structure. Also, many of the routines we will need to help us handle it, are already located in the ROM.

Let's try a simple print using RST 10h. Enter Listing 1 from the TS1000 area above and run it with Rand Usr address.

You should get Error 5 on running this one. Notice how the bottom line is printed and scrolled. Probably not what you expected. We could call the channel open routine to fix this, but there is an upper/lower screen flag that can be temporarily set. If we reset bit 0 of TVFlag, we can print to the upper screen. Insert as the first two instructions:

```
213C5C          Ld HL,TVFlag    ;get TVFlag address
3600            Ld (HL),00      ;reset flag
```

Now run the routine. Works great! A much better way is to only affect the bit needed. This requires the instruction Res 0,(IY+02), which we have not learned yet. You could also have

achieved the same result with the first routine if you ran it with Print Usr address. This sometimes leads to undesireable results, therefore, we will always use Rand Usr address or Let x= Usr address.

Lets attempt to poke a character directly onto the screen. It cannot be done in one easy step as was the case with the TS1000. We must now resort to a complicated routine such as:

Listing 5

```
C630      MakeC   Add A,30h        ;offset to make number
                                   ;a printable character
ED4B365C  Print   Ld BC,(chars)    ;find character table
E5                Push HL          ;save character location
2600              Ld H,00h         ;transfer character
6F                Ld L,A           ;to HL
29                Add HL,HL        ;multiply by 8
29                Add HL,HL
29                Add HL,HL

09                Add HL,BC        ;get offset to character
                                   ;data in table
EB                EX DE,HL         ;address of data to DE
2AB05C            Ld HL,(Store)    ;we are storing address
                                   ;in D-File to print at
                                   ;in store
0608              Ld B,08h         ;# of lines/character
1A        Loop    Ld A,(DE)        ;get pixel data
77                Ld (HL),A        ;poke it to D-File
24                Inc H            ;adjust print pointer
13                Inc DE           ;adjust data pointer
10FA              DJNZ Loop        ;are we done? loop back
                                   ;if not, to complete
21B05C            Ld HL,Store      ;get and adjust print
34                Inc (HL)         ;position
E1                Pop HL           ;retrieve char location
C9        Done    Ret              ;one character printed
```

This routine is worthless without some data to print and another routine to set-up the registers and call it. Notice that there are two entry points. Print is the normal entry, however, MakeC is used to print a number without having its character code (as in raw data, instead of text). The unused location of 5CB0h in the system variables area stores the address of the next print position.

Upon entry to Print, we need to have the character to be printed in A. The HL register points to the character to print and needs to be preserved while Print is executing. Also, note that the program expects the data string to end with a byte containing FFh. Enter the following routine to set-up the registers and Call Print for a test. Your command to execute is Rand Usr Entry.

Listing 6

```
210040    Entry   Ld HL,4000h      ;1st address to print
22B05C            Ld (Store),HL    ;at into our variable
213075            Ld HL,Data       ;data string address
                                   ;assumed to be 7530h
7E        Loop    Ld A,(HL)        ;get character
FEFF              CP FFh           ;is it the end of
                                   ;string yet?
C8        Exit    Ret Z            ;ret if so
CD0080            Call Print       ;else go print it
                                   ;Print assumed to be
                                   ;at 8000h
23                Inc HL           ;advance char pointer
18F6              Jr Loop          ;get next character
```

And here is the data as a hex dump:

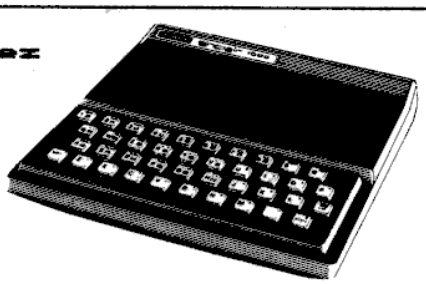
Hex Dump 2

```
54 68 69 73 20 72 6F 75 74 69 6E 65 20 77 69 6C
6C 20 6F 6E 6C 79 20 70 72 69 6E 74 20 69 6E 20
6F 6E 65 20 74 68 69 72 64 20 6F 66 20 74 68 65
20 73 63 72 65 65 6E 20 61 74 20 61 6E 79 20 20
74 69 6D 65 2E FF
```

Be sure you have used the same addresses or change them to suit. If any address is not correct, you may crash.

Well, that's all folks, See ya next time. Syd Wyncoop, 2107 SE 155th St., Portland, OR 97233.

PROGRAMMING CONCEPTS by Albert F. Rodriguez



The following article deals with programming on a Sinclair ZX81 (or TS1000). It is just a portion of a large document, with the remainder to be published in the next issue of TDM, along with a program listing. The listing is a game program, "ZX81 TIC-TAC-TOE", which will serve as the chief example and will be discussed extensively. If readers would like to get a "head start", a complete listing of the program, declarations and array content are available for $6.20 ppd.; or a cassette is available (non-listable) for those who don't want to key in the program, from the author. Albert F. Rodriguez, 1605 Pennsylvania Ave. #204, Miami Beach, FL 33139. (Foreign buyers add $2 for the cassette, or $1 for the listing).

This article was meant to be an exercise in learning about the actual aspects, functions and limitations of the Sinclair ZX81 microcomputer and it's version of the BASIC language. It also provides a tool for finding if, given the technology of this machine, quality software could be produced, and if indeed so, how would someone do it.

First of all, a word of caution to the reader about what lies ahead is in order. Some of the concepts presented in this article, which pertain to programming habits and techniques, were borrowed from works by other parties. Careful attempts in this article have been taken to clearly credit their ideas to each of them. However, the integration of these ideas into a concise whole, as well as any derivations of new ideas thereof, is only this writer, who does accept full credit and responsibility for this work.

As a novice I must say that I was oblivious to the most basic aspects of writing a program using the ZX81: Space and Time. Both of these elements are usually, but not always, directly proportional. The less available programmable memory used, the less time it should take for a program to either execute or transfer from cassette to the computer.

The load time for my program ZX81 TIC-TAC-TOE (which we will look at later on) is 4.17 minutes, without any special hardware assistance. It is saved, self-runs and executes in FAST mode. The actual RAM occupied by the program is 10535 bytes. How this figure was determined reveals a relevant proceedure about programming with this machine.

Both the ZX81 and the TS1000's User Manual fail to discuss how a given amount of RAM programmable memory can be measured in order to determine the total amount of it available, how much is used, and how much remaining for programming.

It is well known that a byte is a memory location that, when active, is holding either blank space, numerical or alpha-numerical characters. Given as an example, a machine with 16k of RAM that is turned on, and is not exhibiting a program, plus the fact that 1k is equal to 1024 bytes, we would expect that it's total available programmable memory is (16 X 1024)...16384 bytes. Yet the display file constantly occupies 1k of memory, even when it is providing nothing but a blank screen with a cursor (for this piece of info, I am grateful to Memotech Corp., Denver, CO 80227). Hence, a blank screen actually represents 1024 bytes of memory occupied by character spaces.

If the programmable area of the screen consists of only 23 lines and 32 columns, then there are only (23 X 32) 736 programmable memory locations (i.e.; only 736 programmable bytes) available in the display file. The other (1024-736) 288 bytes in the display file are found starting with the first line after the cursor line in the work space area and consist of (288/32) 9 lines each having 32 columns. If then, we subtract 288 from 16384 we obtain the total available bytes--16096--that can be used before the machine's entire programmable memory becomes full.

To determine the amount of programmable memory used by a program, the following expression should be placed as the last line of a program and executed with a GOTO from the immediate mode:

```
9999 PRINT PEEK (16404)+PEEK (16405)*256-16597;" BYTES"
```

(In part, for this expression I am grateful to Ms. Toni Baker, "Mastering Machine Code On Your ZX81", Reston Publishing Co., 1981, page 30.) The figure 16597 was determined by taking the starting address of the program area (16509) then adding to it the bytes occupied by this expression (66) and 22 more bytes occupied by another expression (described below) that reveals the amount of programmable memory remaining.

In determining how many bytes of programmable memory remains during, or after a program is being (or was) composed...the following expression should be placed as the first line of a program and executed from the immediate mode by the statement "PRINT USR 16514":

```
1 REM E0RNDLN ACS : FAST AT 5 = GOSUB PI FAST AT TAN
```

(For this expression I am grateful to Dr. Ian Logan, "Bytes Re-

maining--8k", SYNTAX, Vol.2 No.12, Dec. 1981, page 5). With this expression I obtained 5559 bytes of memory remaining after finishing my ZX81 TIC-TAC-TOE program. My memory used (10535) plus my memory remaining (5559) equals 16094. If, however, we add 2 bytes occupied by the cursor sign, then we have the exact sum (16096) of the total available programmable memory for a 16k machine that was mentioned earlier.

Another pertinent phenomenon I realized about both time and space, while writing my program, was about how the ZX81 uses it's available memory to store and manipulate data and the effect of the same on processing time.

My primary concern while writing my program was whether I could fit it all within 16k of RAM. I strived to make my routines as concise and as frugal as possible with respect to the utilization of memory. Consequently, my programming impetus was to make a program that was primarily a space maximizer: "Do all that you have to do, with what you have, in the least amount of space that you can."

It was the inadvertent pursuit of this maxim and the fact that I happened to be reading Chapter 3 of Mr. Randle Hurley's book, "The Sinclair ZX81: Programming For Real Applications" (available from Time Designs), that made me become aware of the fact that less space utilization does not always result in less processing time required for a program or routine to execute properly.

It would be redundant (and beyond the scope of this article) to reiterate the fine explanation that Mr. Hurley provides in his book, as proof, of why the ZX81 sacrifices processing time at the expense of storing and manipulating more accurate data, than say it's forebear the ZX80. Its suffice to say that the ZX81 pays for these new capabilities in terms of more time and less speed during processing.

It should, therefore, be a priority of ZX81/TS1000 programmers, who use BASIC, to accomplish their programming objectives by using instruction-code techniques that minimize memory consumption while simultaneously mzximizing (or at least not minimizing) processing speed.

In his book, Mr. Hurley identifies five important techniques that were helpful toward obtaining an optimum memory-cost-to-processing-speed (MC/PS) ratio when I wrote my program. Though I did use most of these techniques in writing my program, I cannot honestly claim that each part of my program represents the most optimal acheivement of said ratio. Nonetheless, they were employed enough to make somewhat of a positive effect on processing speed and memory savings.

First, there is an advantage in using numerical variable names versus using numerical constants within code expressions. The savings in memory costs can be as much as 40% while still maintaining a faster processing speed. It is for these reasons that all of my subroutines, within my ZX TIC-TAC-TOE program, are called via a respective variable name. This technique also helps to easily access a particular routine in the program for purposes of editing or debugging.

Second, one of the most important aspects about my program is that it requires and makes plenty of use of FOR/NEXT loops. How these loops are structured can have a significant effect on the MC/PS ratio. how they can be made to optimize this ratio is a matter of determining the instruction-code placed inside this kind of loop structure, which will use the least amount of memory to accomplish the necessary accurate manipulation of data with the least amount of loop repetitions. Specific examples that illustrate this principle can be found in Mr. Hurley's book on page 36.

Third, another benefit regarding how to preserve processing speep while saving memory is in how values are calculated with FOR/NEXT loops containing different arithmetic functions. For example, values determined by power functions tend to take longer to compute than if either multiplication or addition were used instead (Ibid., page 36).

Fourth, a substantial optimization of the MC/PS ratio results by initializing numerical variable name constants, in the immediate mode, versus listing them within the programming area. For both memory costs and execution time are augmented when these variables, unnecessarily occupy space both in the instruction area and the variable store. If entered via the immediate mode, they will still be saved on tape; so, why list them in the instruction area again?

Finally, Mr. Hurley's book briefly mentions that where a subroutine is located in a program can affect the MC/PS ratio (Ibid., page 37). Routines (that differ only by size of their line numbers) at the beginning of a program use less memory, but are slower in executing; whereas, those further down the program execute faster but use more memory.

More programming concepts next time.

ZX81 Data Acquisition Module

By

Tim Stoddard

For a long time now I have shied away from using or designing circuits that use A/D converters. I've always figured that they were too difficult not only to build, but also align and set up. So in this issue I've decided to take a stab at an A/D converter for the TS/ZX.

I picked the TLC548 as the work-horse for a few reasons: it's cheap ($6.95), it's easily obtained (Radio Shack #276-1796), it's fairly fast for an inexpensive converter (8 micro-seconds conversion typical), it's CMOS, and it looks quite easy to interface to the TS/ZX. (I found out later on that it's easier to interface than I thought! The entire circuit only took me three hours to set-up, from design to working prototype.) The technical info that comes with the TLC548 really helped to speed the design. This circuit is based on the one in that tech sheet, although re-designed for the TS/ZX. The software is again based on the example in the tech sheet but re-written for the re-designed hardware.

The Sinclair's architecture for I/O is rather complex, ironically, because Mr. Sinclair was trying to keep the hardware design simple! The Z80A used in the TS/ZX is capable of 256 I/O ports using address bits 0-7. However, in order to keep circuit parts count down, the Sinclair's I/O is only PARTIALLY decoded. That means instead of decoding all 8 I/O address bits to come up with an address to read the keyboard, Sinclair only used bit 0 to select the keyboard whenever it is a logical zero, and IGNORES ALL OTHER BITS. So, any EVEN I/O address (bit 0=logic zero) will read the keyboard. That eliminates using all even addresses for external devices. Bit 1 is also used within the computer, when it's a logical zero, so we can't use that bit either. The TS2040 printer is also partially decoded by selecting bit 2=zero and bit 7=one, so if we keep bit 7=zero, then we can use bit 2. The 2050 modem is completely decoded and uses ports 73 and 77 hex. You'll note that they keep bit 7=zero and bits 0 & 1=one to avoid selecting I/O operations in the computer or in the printer. In our A/D converter we will use port 67 hex which will avoid conflict with any of the common peripherals.

You can make this project simple or complex depending on the options you decide to use. ICs 8-19 are optional. To wire up just one channel all you need are ICs 1-7. For two converters just add IC8, and wire up the second half of IC5. You can continue to add A/D converters as shown in the schematic, up to a total of eight. The pin connections on the left side of IC7 are common to all A/D converters that you connect up. For multiple converters, this circuit is more complex than using an ADC809 converter from National Semiconductor, but that converter takes 10 times longer for each conversion than does the TLC548. So for speed's sake the slightly more complex circuit is worth it.

Looking at the schematic #1, IC1 & IC2 serve to decode the I/O address and form the proper read and write signals we will need to access the A/D converters. IC5 is used as a simple output port to "select" the A/D

Continued Next Page.

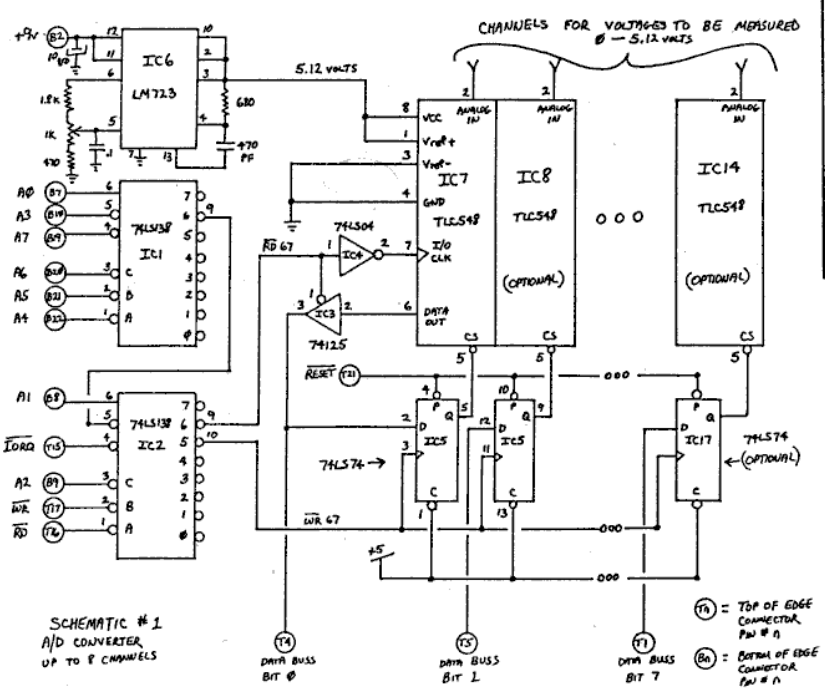
converter that we want to read. Note that upon power up or reset IC5 will be preset, deselecting the attached A/D converter. IC7-IC14 are the converters. IC3 is a tri-state buffer used to read the serial conversion data from the TLC548s. IC6 is the A/D converter supply and reference.

Schematic #2 shows how you can use the other port commands that happen to decode from IC2 for a general purpose I/O port. IC18 can be added if you want a general purpose input port (input 63H) and IC19 for a general purpose output port (out 63H). You could also use that port for any other project you may have in mind. If you can wait until next issue, my next article will be using port 63H for a D/A converter.

To begin an operation, we first select one of the converters by writing port 67H (the "H" designates hex) with the appropriate bit=zero and all other data bits set high. For instance, to select converter #0 we write port 67H with FEH (binary = 1111 1110). So, to select converter #1 we would write port 67H with FDH (binary = 1111 1101). After we have selected one of the converters we then can read the PREVIOUS conversion results from it in SERIAL form. [So after selection, data bit 7 of the result is available from the data out pin (pin 6).] All we need to do is issue a read port 67H instruction and IC3 will put the serial bit on to bit 0 of the data buss for the Z80A to input. IC4 inverts the read port 67H signal so that at the END of the read port instruction the low-going signal will cause the TLC548 to send the next bit out by toggling the I/O CLOCK pin (pin 7). All we need to do in the software is READ PORT 67H, SHIFT LEFT, READ PORT 67H, SHIFT LEFT, ect., until all 8 bits have been read. Then, at the end of the operation, we de-select the A/D by writing port 67H with FFH (binary = 1111 1111). You can deviate from the circuit when selecting a regulator for IC6. I used a 723 because I have a lot of them, and they are readily available. The overall accuracy of the converter is based on this regulator since it supplies the reference voltage to the converter. You could also leave the regulator out and use the regular five volt supply, but accuracy would suffer. The TLC548 is an 8-bit converter. This means that it can "dissect" the voltage being measured by 256, and this would be the smallest measurable voltage. For instance, if the reference voltage is 2.56 volts, then the maximum measurable voltage would be 2.56 volts and the smallest would be 2.56/256 or .01 volts. This would be an ideal reference because each count would be .01 volts. However, the tech sheet indicates that "unadjusted errors may increase if the differential voltage Vref+ to Vref- falls below 4.75 volts". So we need something greater than 2.56 volts to keep accuracy up. If we use 5 volts this would give us the smallest measurable voltage 5/256 or .01953125 volts---a rather hard number to work with. Looking again at the tech sheet you'll see we can supply up to 6 volts in normal operation. So if we use 5.12 volts this would not be too high for the other logic and it would give us 5.12/256 or .02 volts as the smallest measurable voltage---an easy number to work with. Why use the 5.12 as the supply voltage too? If we use the TS/ZX 5 volt supply to supply VCC to the TLC548 and the 5.12 volts as reference voltage only, we would be violating one of the specs for the TLC548. Page 4 of the sheet, the line that states "Positive reference voltage, Vref+ (see Note 2)", this states that the max Vref+ that can be supplied is VCC + .1 volts. We would be .12 volts over VCC.

The following ML routine can be poked into the first REM statement and will give you the PREVIOUS conversion results each time it is accessed. The routine uses relative jumps so it can be placed anywhere. Just remember to add 6 to the starting address of the routine to obtain the address to change the converter selection byte.

```
        LD   B,08H      06 08     06  08    ;initialize counter
        LD   C,0        0E 00     14  00    ;clear result register
        LD   A,FEH      3E FE     62  254   ;select converter
        OUT  67H,A      D3 67     211 103   ;via out instruction
LOOP:   RLC  C          CB 01     203 01    ;shift result register left
        IN   A,67H      DB 67     219 103   ;read converter bit
        AND  A          E6 01     230 01    ;mask off bit 0
        OR   C          B1        177       ;or result register to new bit
        LD   C,A        4F        79        ;put new result into result
register
        DEC  B          05        05        ;decrement bit count
        JR   NZ,LOOP    20 F5     32  245   ;if bit count>0 continue
        LD   A,FFH      3E FF     62  255   ;de-select all converters
        OUT  67H,A      D3 67     211 103   ;via out 67 instruction
        RET             C9                  ;return to BASIC
```



SCHEMATIC #1
A/D CONVERTER
UP TO 8 CHANNELS

Grey & Clifford

COMPUTER PRODUCTS

Ed Grey
(213) 759-7406
SALES AND INFO.

Dave Clifford
(213) 516-6648
TECHNICAL INFO.

P.O. BOX #2186
INGLEWOOD, CA 90305

LARKEN
DISK DRIVE
Controller Boards for
ZX-81
2068

Features Common to Both:
- LDOS on Eprom supports Basic Programs, Code blocks and Arrays
- Commands are-DIRECTORY,LOAD,SAVE,FORMAT, DELETE and EXIT
- Up to 52 files per disk - Directory is DOS maintained
- Can be modified to work on either computer (eprom and cable req)
- The most Powerful DOS for the ZX-81 : Very User Friendly
- 2068 version is Spectrum Emulator Compatible: Extra commands
- Requires double sided 5.25" drive 60 day money back guarantee

Prices : $95.00 (US) for single drive $99.00 for 1 or2 drives
$ 4.00 for drive cable (all other cables included)
include $5.00 shipping : Send certified cheque or M.O.
To - LARKEN ELECTRONICS
(specify ZX-81 or 2068) RR#2 NAVAN ONTARIO
CANADA K4B-1H9

The following is a BASIC program that shows usage of the ML routine above.

```
1 REM 123456789012345678901234
1000 PRINT AT 12,0;"AD1=";AT 12,16;"AD2=";
1004 POKE 16519,254
1005 LET A=USR 16514*.02
1006 POKE 16519,253
1007 LET B=USR 16514*.02
1010 PRINT AT 12,5;A;AT 12,21;B;
1020 GOTO 1004
```

Poke the assembly language routine starting at 16514 which is in the REM statement. Then run the program and it will continiuously read conversions from both A/Ds.

Lines 1005 and 1007 show the usage of the routine, simply call the routine as a USR funtion and multiply the count returned by .02. Of course, if you decide on a different reference voltage, then you would multiply by the result of this formula: Vref+/256.

Location 16519 contains the converter selection byte layed out as follows:

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CONVERTER #	7	6	5	4	3	2	1	0

YOU MUST SELECT ONLY ONE CONVERTER AT A TIME! A ZERO in the bit location selects that converter, and a ONE deselects it. There should be only ONE ZERO in the byte at location 16519 when selecting a converter. This selection method allows up to eight converters to be used while only using one I/O port!

Layout is not critical, however, use a large enough perf board to allow for expansion. My next article will include the addition of a D/A converter to this board, using port 63H. I used a Vector P178-1 tool and the heat vaporizing wire to wire mine up on a perf board, and using one of the Zebra Systems Inc. edge connectors. Hot glue works great for attaching the IC sockets and edge connector to the perf board.

Most of the parts can be obtained at Radio Shack, Jameco, or JDR Microdevices who all advertise in most of the electronic journals. The Zebra edge connectors are $5.95 each (Zebra Systems, 78-06 Jamaica Avenue, Woodhaven, NY 11421), or you can cut a 50 pin wire-wrap edge connector down to the 46 pins required using a fine saw or small cutting disc attachment on a Dremel Moto-tool.

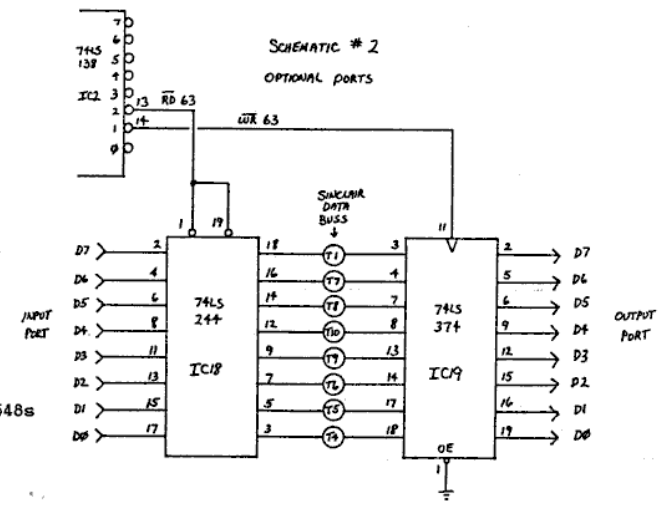
If you have any problems, or even if you don't, drop me a line and let me know how you made out. Tim Stoddard, 85-48 66th Road, Rego Park, New York 11374. CompuServe ID# 73127,2664.

P A R T S L I S T

IC1, IC2	74LS138
IC3	74125
IC4	74LS04
IC5	74LS74
IC6	LM723
IC7	TLC548

O P T I O N A L P A R T S

IC8-IC14	TLC548	
IC15-IC17	74LS74	You'll need 1 for every 2 TLC548s
IC18	74LS244	
IC19	74LS374	



TS 1000/1500 PROGRAM CHAINING

PART FIVE

BY

Earl V. Dunnington

In the previous episode of this series, the use of an endless tape with a chained program was presented. I failed to mention that the tape recorder could be left running while the program is being used. The maximum access time for the next module would be one minute for the one minute tape or three minutes for the three minute tape. Due to the manner in which these tapes operate, just how long it would take before the tape failed, when used in your own recorder, is a matter for you to investigate. The best method of using a chained program on endless tape, if your tape recorder has a jack for an external motor switch, is to leave the recorder on play and use a foot switch (Radio Shack #44-610 $2.99) to turn the motor on and off. This same method could also be used, if the entire program requires more than a three minute tape or you do not wish to use an endless tape, by recording the modules several times on a regular tape. Using the demonstration program as an example, you would record the modules as follows: RT, TE, PRT, TE, PRT, TE, PRT, ect.

Even those readers with only a 2k RAM, now have a basis for a full-featured word processor, a file program or a data base by adding additional modules. One of the first would probably be a SAVE/LOAD module. Because the data (text) is stored above RAMTOP and BASIC on this computer does not have a command to SAVE or LOAD data from this area, a machine code program is required. Unfortunately there are only so many ways to write a program like this and they have been published and copyrighted, precluding me from presenting a listing in this article. A data SAVE/LOAD program was published August 1984 in the now defunct SYNTAX. A similar program is in David B. Wood's book "Machine Language Programming On The Timex/Sinclair" from Siriusware. For thoe readers who have a ZX81 or TS1000 with at least 16k RAM and who do not mind losing some storage space above RAMTOP, as the program itself occupies 2k RAM, there is a faster-

than-normal SAVE/LOAD from anywhere in memory program available from Zebra Systems Inc. (Z-XLR8 Cat# GR04 $11 plus $3 for postage). According to Zebra, this program will not work on the TS1500. Perhaps one of our more accomplished Z80 Assembly Language programmers could develop some pokes to this program for the TS1500.

Additional modules you might wish to add to the chained program could be Text Corrections, Cut and Paste, Right Justification, Letter Heading, and Search modules. As the tree modules presented in the previous episode were developed strictly for demonstration purposes, feel free to use them, as is, or changed in your own program. In order to do this, you should know how the value for RAMTOP and the variable B were determined.

For a more detailed explanation of the procedure involved, please see "Adventures In The RAM Jungle" (Sept/Oct '85 to Jan/Feb '86 issues of TDM). This method for determining the Upper and Lower Limits of the Safe Area will not work when there are expanding un-dimensioned strings in the program, due to the random use of the Spare Area of the RAM by the ROM routine. The Wicked Wizard of ROM strikes again!

The Spare Area of the user-available RAM is located between the top of the Calculator Stack and the bottom of the Machine Stack (see page 128 of the TS1000 or page 154 of the TS1500 User Manuals). When the computer is turned on, the addresses in this area are filled with zeros. Upon loading and running a program from tape, the Calculator Stack expands upwards and the Machine Stack downwards into the Spare Area leaving, what I call the Safe area untouched. The Upper Limit of the Safe Area is easily determined, as the Machine Stack leaves a trail of values other than zeros as it contracts upwards. Determining the Lower Limit is more difficult, because as the Calculator Stack contracts downwards it either replaces the zeros when RAMTOP is set above address 19711 or fills the addresses with the values stored in the Safe Area just above the Lower Limit, if RAMTOP is set to 19711 or less, an INPUT command is in the program, and the DELETE command is used to correct the input. The Wicked Wizard does not play by the same rules all of the time! In addition, when RAMTOP is set to 19711 or less, the computer is in the minimum display file mode and as characters are printed on the screen, the display file expands, pushing the Calculator stack upwards. Therefore, in the case of the demonstration program, we cannot allow the program to be listed when determining the Lower Limit of the Safe Area.

The number of bytes in the Safe Area plus 36, of the module requiring the most memory in order to operate determines the address to which we can lower RAMTOP and still have both the entire program and the computer operate properly.

I must admit that having a TS1000, I set RAMTOP to 2k (address 18432) and developed the TE module first, as I expected it to require most memory. The text in this case being POKEd to 18433 and above. I then determined the Safe Area for that module. Next I programmed the PRT module, also with RAMTOP set to 2k, and determined the Safe Area for it. Then I programmed the RT module so that it would not move RAMTOP from 18432, as otherwise the Safe Area for that module could not be determined. However, in order that those readers with only 2k RAM can follow the method used in determining the Safe Areas, we will work with RAMTOP set to 1k or address 17408.

```
  1 REM 1234
 10 SAVE "RT"
 20 FAST
 30 POKE VAL "16388",VAL "000"
 40 POKE VAL "16389",VAL "68"
 50 POKE VAL "17407",VAL "62"
 60 POKE VAL "17406",VAL "0"
 70 POKE VAL "17405",VAL "6"
 80 POKE VAL "17404",VAL "118"
 90 POKE VAL "16386",VAL "252"
100 POKE VAL "16387",VAL "67"
110 RAND USR VAL "16514"
120 GOSUB VAL "140"
130 LOAD "TE"
140 RETURN
141 POKE 16514,49
142 POKE 16515,252
143 POKE 16516,67
144 POKE 16517,201
```

ROUTINE TO SET RAMTOP TO 17408

FIGURE NO. 4

Here is some homework to be done in preparation for the conclusion of this series. You will need a tape that is blank on at least one side. Turn on your computer and set RAMTOP to address 17408 by entering the following immediate commands:

```
POKE 16388,0
POKE 16389,68
NEW
```

To check that RAMTOP has actually been moved, enter the direct command:

```
PRINT PEEK 17407
```

The result should be a 62 printed on the screen.

Load the TE module from your master tape, your operating tape, or type in the TE module from the listing (Fig.2) on page 16 of the Nov/Dec '86 issue of TDM. Change line 180 to read:

```
180 LET A=VAL "17409"
```

Enter the direct commands:

```
CLEAR
DIM T$(32)
LET B=18401
```

The variable B used here was computed for 2k RAM, but can be used with larger RAM for the purpose of determining the Safe Areas. After checking the program against the listing, record the TE module on tape using the direct command:

```
GOTO 140
```

To clear the computer enter NEW and either load the PRT module from your master tape, your operating tape, or type in the listing (Fig.3) on page 16 of the last issue. Change line 140 to read:

```
140 FOR N=VAL "17409" TO B
```

Enter the direct commands:

```
CLEAR
DIM A$(1)
LET B=18401
```

After checking against the listing, record the PRT module, as the next program on the tape, using the direct command:

```
GOTO 10
```

To clear the computer enter NEW. The RT module must not be loaded from the operating tape, as it would change RAMTOP. It must be revised so that it will not move RAMTOP from 17408 in order to determine the Safe Area of this module. If you recorded it along with lines 141 to 144 on a master tape, you can load it into the computer without running it. Otherwise, type in the listing from page 16 (Fig.1) of the last issue. See if you can make the changes to set RAMTOP to 17408 without referring to Fig.4 in this issue. Check the program very carefully against Fig.4 and then use the direct command:

```
GOTO 141
```

To poke the machine code into the REM statement, delete lines 141 to 144 , as they are no longer needed to save memory. Record the module as the next program on the tape, using the direct command:

```
GOTO 10
```

When the diagonal LOAD lines appear, use the BREAK key and stop the tape. Enter the direct command:

```
PRINT PEEK 16388+256*PEEK 16389
```

If all is well 17408 will be printed on the screen.

Label these recordings as preliminary. They will be used to demonstrate the method of determining the Safe Area of each module and the minimum address to which RAMTOP can be set. This will be covered in the conclusion of this series of articles.

Happy Letters

A Teaching Tool For Young Children Using Sound(!) And Graphics With The ZX81/TS 1000/1500

By Gale Hensley

This program generates a random character, either a letter or a number, and displays it on the screen, enlarged to eight lines. The Player presses the matching key on the ZX81 keyboard and is rewarded with a "Happy Face" and a song. A wrong response gets a "Sad Face" and a beep. You also get a beep each time a key is pressed. The program is suitable for teaching young children the keyboard. My daughter, aged 2½ years, finds it very entertaining and has mastered the alphabet and numbers on the keyboard. It required very little encouragement, just a willingness to load and supervise her play.

To play, you need a ZX81 or TS 1000 with at least 2k, and an external amplifier (example: RS Mini Amp Cat. #277-1008B). A "real" keyboard helps, but is not required. Young children have much better results on a keyboard than on a Sinclair membrane.

The sound is generated by the MC routine in line zero. This routine is from a program by Neal Bridges, which appeared in TS USER, Vol.1 Nov.7. You must be in FAST mode when calling this code or it won't work. Sound output is via the microphone jack, and frequency is controlled by the values POKED in locations 16519 and 16520. Duration of the tone is controlled by a delay loop (see lines 605 to 640).

To enter the program, first enter lines 700 to 750. Then enter a REM statement (line 1) with 38 characters. RUN and list the program. If it looks OK, then POKE 16510,0 to change the line number to zero. SAVE and then test as follows: First hook up the amp to the MIC output from the computer. Turn on the amp and turn up the volume until you hear the 60hz hum. (NOTE: you can also hear the music without the amp by turning up the TV volume very loud or placing a transistor radio near the computer, tuned to a very weak station.) Second, FAST, followed by RAND USR 16514 should produce a short beep and a horizontal pattern of bars across the screen similar to the loading pattern. The error code should return 0/0. Now, enter the rest of the program, lines 10 to 690. SAVE it...and then RUN. Correct any errors, delete lines 700 to 750, and SAVE the final version.

One final tip is in order. You can break into the program by pressing the SPACE key. I had to make a cardboard space guard to keep my daughter from breaking out every few minutes. It seems that getting Daddy to fix the 'puter is almost as much fun as her "Happy Face" program.

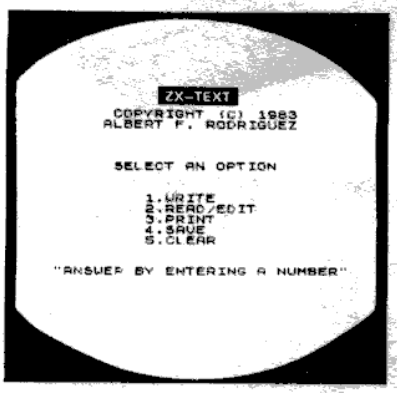
I can make a copy of the program for anyone who'll send a good quality tape, and a S.A.S.E. to: 4411 West 2nd St., Amarillo, TX 79106.

```
MC LOADER FOR HAPPY LETTERS

**********************************

FIRST MAKE A REM WITH 38 SPACES
THEN GOTO 700
THEN TEST RUN THE PROGRAM
THEN DELETE 700 - 750

**********************************

 700 LET A$="260A2E01010220C5D3F
FCD9C40C1DBFECD9C402D20EE2520E9C
90405280210FE0C0DC80D18F8"
 705 LET B=16514
 710 FOR N=1 TO LEN A$ STEP 2
 720 LET A=(CODE A$(N)-28)*16+CO
DE A$(N+1)-28
 730 POKE B,A
 740 LET B=B+1
 750 NEXT N
```

```
   0 REM A"I"  "4VAL PEEK COPY LN
 RNDAT <= RETURN LN RNDH4 INPU
T 94 DIM TAN  C"( RETURN £$COS
$/ SAVE
  10 RAND
  20 FAST
  25 CLS
  30 LET A$=CHR$ (INT (RND*36)+2
8)
  70 LET C=CODE A$
  80 FOR H=0 TO 7
  90 LET P=PEEK (7680+C*8+H)
 100 LET V=128
 110 FOR G=0 TO 7
 120 IF P<V THEN GOTO 150
 130 PRINT AT H,G;"■"
 140 LET P=P-V
 150 LET V=V/2
 160 NEXT G
 170 NEXT H
 180 SLOW
 220 IF INKEY$="" THEN GOTO 220
 222 LET X=CODE INKEY$
 223 FAST
 224 LET Y=USR 16514
 225 IF INKEY$<>"" THEN GOTO 225
 230 FOR H=0 TO 7
 235 LET P=PEEK (7680+X*8+H)
 240 LET V=128
 245 FOR G=0 TO 7
 250 IF P<V THEN GOTO 265
 255 PRINT AT H,10+G;"■"
 260 LET P=P-V
 265 LET V=V/2
 270 NEXT G
 275 NEXT H
 300 IF X=C THEN GOTO 400
 302 GOSUB 500
 305 FOR H=0 TO 7
 310 PRINT AT H,10;"          "
 315 NEXT H
 330 GOTO 180
 410 PRINT AT 13,10;"    "
 420 PRINT AT 14,10;"  ■  ■  "
 430 PRINT AT 15,10;"  ■  ■  "
 440 PRINT AT 16,10;"  ■  ■  "
 450 PRINT AT 17,10;"  ■■■■  "
 460 PRINT AT 18,10;"    "
 470 GOSUB 600
 475 SLOW
 480 PAUSE 200
 490 GOTO 20
 500 PRINT AT 13,10;"    "
 510 PRINT AT 14,10;"  ■  ■  "
 520 PRINT AT 15,10;"  ■  ■  "
 530 PRINT AT 16,10;"  ■  ■  "
 540 PRINT AT 17,10;"  ■■■■  "
 550 PRINT AT 18,10;"    "
 551 FAST
 552 POKE 16519,3
 554 LET X=USR 16514
 555 POKE 16519,2
 558 SLOW
 560 PAUSE 100
 570 RETURN
 600 FAST
 605 FOR M=1 TO 3
 610 FOR N=1 TO 5
 620 LET X=USR 16514
 630 NEXT N
 640 NEXT M
 645 POKE 16519,2
 650 POKE 16520,176
 655 FOR N=1 TO 15
 660 LET X=USR 16514
 665 NEXT N
 670 POKE 16519,2
 675 POKE 16520,32
 680 SLOW
 690 RETURN
```

POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE FOR ZX81, T/S1000 and T/S1500 COMPUTERS

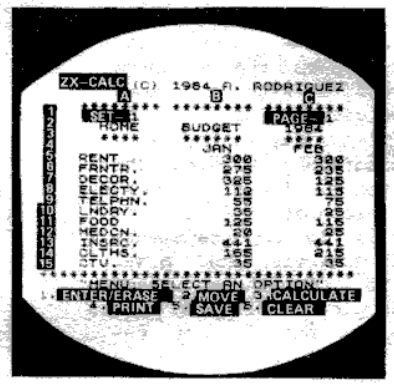
ZX-TEXT



A word processor is to a computer user what a typewriter is to a typist, except that the former has more advantages than the latter. ZX-Text can operate in 16-64K RAM providing from 1300 to 6500 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back-space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer. (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends...etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$16.95

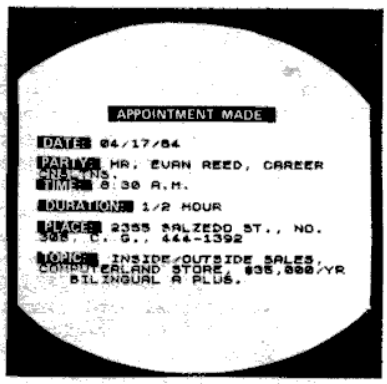
ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters / spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters / cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX-81 / TS1000. It offers a unique *SUM function that totals one or more rows / columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows / colums. Loading of data into more than one cell can occur across / down one or more row / column simultaneously. With vertical windowing you can arrange a set of columns in any order, or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter / erase, move, calculate, print, save and clear the spreadsheet. Enter / erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire sreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute / relative replication, down / across a column / row, is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX / TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial model rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$16.95

ZX-CALENDAR



Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 25 appointments in 16K, 100 in 32K or 180 in 48K and 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search / check / sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently originate, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know, by using it, to never be caught astray by any time-frame.

$16.95

$3.00 SHIPPING AND HANDLING / PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** **FLORIDIANS ADD SALES TAX**

L-1 © 1984

RESOURCES FOR YOUR SINCLAIR
FROM TIME DESIGNS MAGAZINE COMPANY

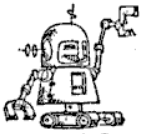
GREAT BOOKS FOR YOUR SINCLAIR

OFFICIAL 2068 TECHNICAL MANUAL.....$25.00

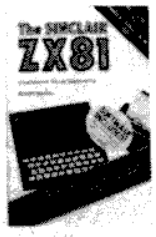
CONTROL THINGS with your Timex/Sinclair by R.L. Swarts.....$5.00

ZX81: Programming for Real Applications by Randle Hurley.....$6.00

Includes 12 Software Programs on a Cassette Tape

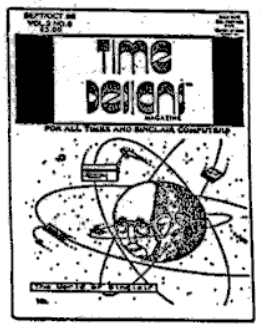


Build a Microcomputer - Controlled Robot by B.C. Taylor.....$17.00



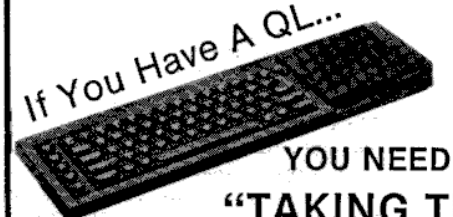
BOTH *"Control Things"* and "ZX81" Package For Only **$10.00**

TIME DESIGNS MAGAZINE



Still Only $15 Year Six Issues

TRY A SUBSCRIPTION TODAY!



YOU NEED

"TAKING THE QUANTUM LEAP"

By Mike de Sosa

The Ultimate Resource For Your Sinclair QL

AVAILABLE SOON!

Published Exclusively By TIME DESIGNS
Write or Call for Information

TIME DESIGNS MAGAZINE BACK ISSUES $3.00 EACH

Nov/Dec 85 VOL.2

Jan/Feb 86

March/April 86

May/June 86

July/August 86

Sept/Oct 86

Nov/Dec 86 VOL. 3

Volume One Issues are no longer available individually (See "BEST OF TDM").

THE BEST OF TIME DESIGNS VOLUME ONE

The Best of TIME DESIGNS Volume 1

$10.00

QTY	ITEM		AMOUNT
	TIME DESIGNS MAGAZINE Subscription - One Year	$15	
	TIME DESIGNS BACK ISSUES Specify Which:	$3.00 each	
	THE BEST OF TDM Vol. 1	$10	
	TS 2068 Technical Manual	$25	
	Control Things With Your Timex Sinclair	$5	
	ZX81: Programming for Real Applications (includes tape)	$6	
	SPECIAL PACKAGE DEAL: Both "Control Things" and ZX81"	$10	
	Build a Microcomputer Controlled Robot	$17	
	Taking The Quantum Leap (Available February 87)		
	WE PAY THE POSTAGE ON ALL ORDERS!	TOTAL	

FOR MAGAZINE SUBSCRIPTION (please circle one below):
NEW SUBSCRIPTION THIS IS A RENEWAL THIS IS A GIFT

SHIP TO:

NAME_____

ADDRESS_____

CITY/STATE_____

ZIP_____

PHONE_____

PLEASE PRINT NEATLY.

CREDIT CARD
(Circle One)
MASTERCARD / VISA

Acct. Number_____

Exp. Date_____

Signature_____

Note: 4% charge added to all Credit Card Orders.

PAYMENT
Check or M/O.

VISA and MASTERCARD ACCEPTED

VISA MasterCard

24 hour ORDER LINE
(503) 824-2658
For Books, Subscriptions and Renewals

time designs

TIME DESIGNS MAGAZINE CO.

29722 Hult Rd., Colton, OR 97017

the CLASSIFIED

FREE ADS FOR SUBSCRIBERS

PEN PALS WANTED: I'm going on Active Duty in the U.S. Navy and I would like pen pals so as to keep up on my "hobby". User Groups individuals, ect. I program TS 1000, 2068, QL, ZX Spectrum also. Also interested in MIDI. Dave Johnson, 2399 St. Rt. 95, Edison, OH 43320.

FOR SALE: QL, Sinclair Monitor and Printer. Memory boards, software, new blank cartridges, various QL books, QL technical manual, spare power supplies, ect. CALL for prices and details. (603) 352-9836.

FOR SALE: Timex Sinclair 1000 computer in Metal Sloping Cabinet, with keyboard, power switch, Westridge modem, Aerco Centronics Parallel Interface, Memotech 64k RAM, RS cassette recorder, large library of software including- QSAVE, Extended Basic, Sincartist, Memotext WP, games. Available as complete package or by piece. If interested, please contact- Richard Beier, 1 Darwin Dr., N. Merrick, NY 11566. Leave Email on CompuServe. ID# 73137,1565.

FOR SALE: ZEBRA one-drive disk system. Comes complete with extension interface board and formatting/demo disk. $200 ppd. UPS. Also Gorilla/Banana full sized Centronics parallel printer with true descender modification chip installed. $100 ppd. UPS. Warren Jackson, 11141 Edgemere Ter., Roscoe, IL 61073. Or leave Email on Compuserve. ID# 73177, 1017.

MONEY MACHINE II starring Banna Brite! All new format. Banna turns the letters. The ultimate word game for the TS2068 from ABBA SOFT. $12 ppd. Herb Bowers, 2588 Woodshire Circle, Chesapeake, VA 23323.

STAR TREK a Strategy Game for the TS2068--7 levels of play. Based on 1973 Public Domain HP2000C version. Send $8.00 to- Michael E. Carver, 1016 NE Tillamook, Portland, OR 97212.

FOR SALE: QL Computer, Sinclair Monitor, QL Modaptor(modem)... all BRAND NEW units in original cartons. ALL FOR $400. Write or call. John Willis, 10531 Galena Star, Dallas, TX 75228. Phone (409) 693-2249.

WANTED: Used TS2068 with power supply; used 2050 modem with power supply. Both good condition and inexpensive. CALL or WRITE: Don Hornberger, 8514 Mathilda #2, St. Louis, MO 63123. Phone- (314) 832-7434.

SINCUS NEWS-$8/yr=6 issues. TS2068 oriented user group news letter- mail check to SINCUS, 1229 Rhodes Rd., Johnson City, NY 13790; non-profit amateur users oriented group.

SOFTWARE for the TS2068, Spectrum, and TS1000. Send for free listing. Price $5.00 and $2.00. All original software. Garvin Bertrand, #311 135 Lynnview Rd. S.E., Calgary, Alberta, Canada T2C 2G6. Tel: (403) 236-4465.

WANTED: Sinclair ZX80 (the original ZX). Must be good condition and working. Please send information and price to- Tim Woods, c/o Time Designs Magazine, 29722 Hult Rd., Colton, OR 97017.

WANTED: Hunter Board and/or other TS1000 "what have you's". Contact- Fred Henn, c/o Rockelman & Henn Pump Co., Inc., 1333 Military Rd, Buffalo, NY 14217.

TRADE: My Parallel Gorilla Banana printer for a good Serial 64 col. printer. Trade: My TS2068 Tasman Parallel Interface for a good Serial printer interface for the 2068. Trade: A very good Mindware TS1000 printer for a TS2040 printer...also have extra TS1000. Timothy J Herman, 100 Gilman St. #13, Verona, WI 53593.

MONEY MACHINE I, $10 ppd. Money Machine II, $12 ppd. On 2068 tape or Aerco 5.25" disk. Send 22-cent stamp for full catalog. Chia-Chi Chao, 73 Sullivan Dr., Moraga, CA 94556-1209.

PROFILE Now On Wafadrive...SAVE & LOAD to drive A/B or tape...Send $9.95 + .50 w/your PROFILE tape or a copy of your sales receipt and a blank tape to: John Deering, 136 Neverbreak Dr., Hendersonville, TN 37075.

BRAND NEW ITEMS FOR SALE: Two (2) TS2068s...$75 each; one (1) TS1500 ...$35; one (1) Sears Personal Typwriter/Printer (Serial)...$60; one (1) rare Sinclair Spectrum specially built for U.S....$75; one (1) Sinclair Microdrive w/ Sinclair Interface One...$75; one (1) Multiface(Romantic Robot) for Spectrum...$35. New Damco Spectrm Rainbow Interfaces for the 2068... $29 each; misc. business software for the wafadrive. Also one used 2040 printer...$15 (no p.s.). Other items. Write for list. Dave Maccarone, 67 Bradley Ct., Fall River, MA 02720.

2068 SOFTWARE: Send S.A.S.E. for FREE CATALOG to- Timeware, Inc., 1907 1/2 W. Genesee St., Syracuse, NY 13204.

BACK ISSUES OF TDM: All Vol.2 issues (6) are available...Nov/Dec 1985 thru Sept/Oct 1986 for $3 each ppd. Index to back issues in Nov/Dec 1986 issue (also available for $3 ppd.). No Vol.1 issues left in stock...but collection of prog.-articles, ect. is available called "The Best of Time Designs Vol.1" for $10 ppd. Send check/mo or credit card info to- Time Designs, 29722 Hult Rd., Colton, OR 97017. Quantities are limited.

Do you have some equipment or a program that you would like to sell? Looking for something hard to find? Place an ad in **THE CLASSIFIEDS!** Subscribers can place one free personal ad in each issue. Ad size is 32 Col. wide (like 2040 paper) and maximum of six lines. For additional lines - $3 each. **NON-SUBSCRIBERS** and **DEALERS:** $4 a line. **DEADLINE FOR ALL CLASSIFIED ADS:** Two weeks before publication date. Mail your ad to: **TIME DESIGNS MAGAZINE, The Classifieds Dept., 29722 Hult Rd., Colton, Oregon 97017.**

WMJ DATA SYSTEMS

Serving the Timex/Sinclair community since 1983

- Free shipping on all prepaid orders!
- Shipment within 24 hours on most of our orders!
- Your personal check is gladly accepted with no delay!
- C.O.D. phone orders accepted! (Shipping charges added)

TS1000

ACZ GENERAL LEDGER	TS1000	$19.95
SUPERTAPE-45 PROGRAMS	TS1000	18.95
TEXTWRITER 1000-32 col.	TS1000	12.95
SPEECH SYNTHESIZER	TS1000	16.95
STOCK PLOT	TS1000	14.00

BOOKS AND ACCESSORIES

KRAFT SYSTEM JOYSTICKS	TS2068	$ 9.95
HACKER'S HANDBOOK	ALL	9.95
BASICS AND BEYOND	TS2068	7.00
C10 CASSETTES	25 for	21.00
C20 CASSETTES	25 for	23.00

TS2068

ACZ GENERAL LEDGER	TS2068	$19.95
THE DEALER'S DEN	TS2068	19.95
MACHINE CODE TUTOR	TS2068	18.50
TOMAHAWK	TS2068	16.95
NIGHT GUNNER	TS2068	16.95
FIGHTER PILOT	TS2068	16.95
CRITICAL MASS	TS2068	15.95
SABOTEUR	TS2068	15.95
BASIC TOOLKIT	TS2068	13.95
MSCRIPT (specify conf.)	TS2068	24.95
DIAMOND MIKE	TS2068	16.95
COLONIZE THE UNIVERSE	TS2068	16.95
STOCK PLOT	TS2068	16.00
All QUICKSILVA Software	TS2068	CALL

WMJ DATA SYSTEMS—TD

4 Butterfly Drive
Hauppauge, NY 11788

(516) 543-5252 (2 p.m. - 5 p.m. Best)

FREE Apple, Atari, Commodore, IBM...Catalog
FREE Timex/Sinclair Catalog - Write or Give us a CALL!

WEYMIL presents...

A small collection of truly innovative products for Sinclair computers

NEW FOR 1987 !!

MULTI-TASKING ON THE TS 1000?

BELIEVE IT !!!

WEYMIL continues to deliver what everyone else says is "impossible." What's the hottest feature in computers today? MULTI-TASKING! Until now, most people thought that you had to buy an expensive 68000 machine for multi-tasking. Please allow us to introduce NOVA 1000, the software that delivers MULTI-TASKING to the TS 1000. No gimmicks, no tricks. A full multi-tasking environment for the TS 1000 at a very affordable price. Our developers have outdone themselves on this one. Please read on.

NOVA 1000 comes complete with these powerful built-in features. An honest-to-goodness switchable real-time clock that runs concurrent with your programs ON THE SCREEN. A switchable automatic line tracer to allow you to follow the operation of your BASIC programs. Switchable auto-repeat on all keys, just the thing for creating long REM statements. There's more.

Each copy of NOVA 1000 comes complete with 3 complete versions. Version 10 contains the complete NOVA 1000 program in statement 0 REM to allow you to incorporate its features in many existing programs or programs of your own design. Version 20 automatically locates itself to the 8 to 16K region for Hunter Board of 64K applications. Version 30 automatically lowers RAMTOP for 16K configurations.

Consider these possible applications of NOVA 1000. A background auto-dialer for BBS applications would allow your computer to be used for other purposes while waiting for access. The real-time display could be used to keep track of BBS time usage, great for pay boards or long distance calls. Multiple M/C programs can now be run simultaneously. The real-time clock can be used as a timer to activate or de-activate several programs at once. BASIC and M/C programs can be run together. Package M/C programs as memory resident BATCH files. Create printer buffers for W/P applications. Transfer data between two or more on-line programs. Control multiple real-time I/O events for powerful low-cost robotics. Monitor the actual running of BASIC programs rather than just watching the results. There's still more.

NOVA 1000 is very user-friendly and well-documented in laymen's language. You would expect a program with these features to be memory intensive, NOVA 1000 uses only 512 bytes! NOVA 1000 does not slow down your computer. SLOW mode programs can actually have speed increases up to 400%! Registered users will receive FREE periodic application update sheets.

We feel confident that NOVA 1000 will redefine the standards for TS 1000 programming. Consider this offer. If you can develop imaginative programming applications utilizing NOVA 1000, we will provide our proven marketing expertise. We will handle advertising, distribution, and customer service PLUS pay you the most competitive rates in the industry. All you have to do is create and we'll take it from there. This offer is restricted to registered users.

NOVA 1000 for the TS 1000 only $ 20.00

TRACER TS2068

The perfect de-bugging tool for BASIC programs. This program allows you to actually see each line of your BASIC program as it is executed. This program prints not only the line number in use, but the commands as well. It uses only the bottom of the screen for display rather than interrupting the screen display. De-bugging has never been easier.

TRACER FOR THE TS 2068 only $ 15.00

SOUNDESIGN TS2068

The best sound development program we've seen for the TS2068. This program allows you to manipulate the sound chip of the TS 2068 in ways never before imagined. The sound effects, synthesis, and music this program can create have to be heard to be believed. Along with built-in sound routines, this software allows you to develop and mix your own as well. Once you have created the sound you want, all you have to do is press a key and you get a print-out of the line of BASIC required for you to implement that sound in your favorite program. This program is not only a powerful utility for any software library, it is also just plain fun.

SOUNDESIGN for the TS 2068 only $ 15.00

SPECIAL OFFER!!!!

Save $ 5.00 when you order the combination of THRUST, RIGTER, JOYSTICK INTERFACE, and KRUNCHER 1000 and pay only $64.95

SHIPPING INSTRUCTIONS: Please enclose $2.00 shipping and handling with your order.

WEYMIL CORPORATION

Box 5904

Bellingham, WA 98227-5904